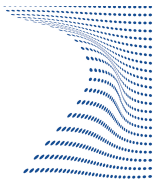
Text miner's little helper: scalable self-tuning methodologies for knowledge exploration


ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR



Doctoral Dissertation
Doctoral Program in Computer and Control Engineering ($31^{st}$ cycle)

# Text miner's little helper: scalable self-tuning methodologies for knowledge exploration

By

**Evelina Di Corso**

******

**Supervisor:**
Prof. Tania Cerquitelli, Supervisor

**Doctoral Examination Committee:**
Prof. Annalisa Appice, Università degli studi di Bari, Italy
Prof. Khalid Belhajjame, University Paris-Dauphine, France
Prof. Elena Maria Baralis, Politecnico di Torino, Italy
Prof. Rosa Meo, Università degli Studi di Torino, Italy
Prof. Genoveva Vargas-Solar, CNRS, France

Politecnico di Torino
2019

# Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Evelina Di Corso
2019

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

*Alla mia mamma,*

*la donna più importante della mia vita,*

*sei la luce che illumina il mio cammino.*

*Gonna rise up, turning mistakes into gold.* (Eddie Vedder)

*I have not failed. I have just found 10,000 ways that won't work.* (Thomas Edison)

*Great minds discuss ideas; average minds discuss events; small minds discuss people.* (Eleanor Roosevelt)

*Mathematics, rightly viewed, possesses not only truth, but supreme beauty – a beauty cold and austere, like that of sculpture, without appeal to any part of our weaker nature, without the gorgeous trappings of painting or music, yet sublimely pure, and capable of a stern perfection such as only the greatest art can show.*
*(Bertrand Russell)*

# Acknowledgements

I express my sincere appreciation to those who have contributed to this thesis and supported me in one way or the other during this amazing journey for without any of them, this research work would not have been possible.

I would like to acknowledge the guidance of my supervisor, Professor Tania Cerquitelli, for providing me an opportunity to complete my PhD thesis. I appreciate the words spent and the time dedicated to make my work stimulating and productive. Her valuable suggestions, comments and advise encourage me to learn more day by day. Despite the difficulties, I appreciate the time that you have given me. Thank you for teaching me how to work hard and how not to wait on others to do what needs to be done.

Besides my advisor, I would like to thank the reviewers of my thesis committee: Prof. Annalisa Appice and Prof. Khalid Belhajjam for their insightful comments and encouragement, but also for the hard questions which led me to widen my research from various perspectives.

I owe thanks to a very special person, the love of my life, Luca for his continuous and wonderful love, support and understanding during my PhD researches and which made possible the completion of this thesis. Thanks for being my eyes when I could not see and being my inspiration when I knew I could do it. You were always been there when I thought it was impossible to continue, you helped me keep things in perspective with your determination and fortitude. I really appreciate your contribution and I deeply appreciate your trust in me. In particular, I dedicate this important brick of my life to you; I hope it will be the basis for a solid life together.

I thank my family. Thank you for teaching me good from wrong and for encouraging me to keep my dreams in sight. This allowed me to become a better person. Thanks for showing me to not let any obstacles stop me and to create a smile from my frown at the worst. Thank you for saying that you care about me and show me just how

special love should be. Thank you for the tears you have dried me when I feel sad and for calming me when mad takes over. These words are not enough to express my gratitude to you. Thank you Pepy for being the best brother in the world. You are my family, because regardless of everything, I will love you forever and always, no matter what, and through the toughest of times.

A family is one of life's greatest blessings, so I consider myself extremely blessed to be a part of two. Dear Lucia and Vito, there is no way I could ever thank you enough for being my second family. You may not be my family by blood, but you are my family by choice. You have given me advice, even when it was hard to hear, and have loved me through every circumstance that came my way. You have smiled with me during the highs of my life and cried with me through the lows. At times in my life, you have been my safe haven, my refuge, and my escape. Without realising it, you have made an unforgettable impact on my life.

I thank all the colleagues and friends at Lab5. You have made these years somewhat enjoyable. However, you will always be my lovely little goats. In chronological order, Alberto and Emanuele, for being my first friends during my PhD. You will always have a special place in my heart. Federica M. and Federica B., who made every moment together special. Federica M., you are simply solar and special. Thank you, Federica B., especially for your friendship, sincere and special. You have been a shoulder in difficult times and a great friend in moments of joy. I will not forget the cries and smiles passed together. I wish you the best that all this world could give. All your dreams can come true if you have the courage to pursue them. I thank Luca V. for his deep thanks in his thesis, and Fabrizio with whom a cup of tea could never be denied. Thank you little Angel, I wanted a graduate student like you. Thanks to Francesco, for making me start using the much-hated Python and thanks to Eliana, Elena, Andrea and Federico for integrating so well into our squadron. You will always be my little children. A special thanks to you Stefano, my partner in adventure on the island that is not there. You are the sincerest person I have ever met, I will never forget you. Thanks also to Alessandro, despite being only a scholarship researcher. Your dynamism will be lacking in this laboratory. And to you, Giuseppe, that you brought joy to this laboratory. I would also like to express my acknowledge to the most beautiful woman in the world, Giulietta, and Mirko, to be a very good worker. Tamer, thank you for your determination and courage, you are an example of life for me.

I also thank all the thesis students who left me a small piece of them in my heart. I will always remember you. Thanks also to the wonderful technicians and secretaries who have literally saved me in many situations. Rita, Gian Piero, Giancarlo, Lino, Franco, Flavio, Anna e Salvatore, you are so special for me. And thanks to those who have always found a little time to go to the Lab5 to find me.

# Abstract

Nowadays, large volumes of heterogeneous data are continuously collected at an ever-increasing rate in various modern applications, ranging from social networks (e.g. Twitter, Facebook), digital libraries (e.g. Wikipedia), smart city environments, Internet of Things (IoT) services and so on. In addition, we are in an age of data-intensive science and we are witnessing the unprecedented generation and sharing of large scientific datasets. Indeed, the pace of data generation has now far exceeded the pace of data analysis.

The analysis of these data collections is challenging, as it is a multi-step process in which the data scientists tackle the complex task of configuring the analytics system to transform data into actionable knowledge to effectively support the decision making process.

A plethora of algorithms are currently available for performing a given data analysis phase, but for each one the specific parameters have to be manually set and the obtained results validated by a domain expert. Moreover, real datasets are also characterised by an inherent sparseness and variable distributions, and their complexity increases with the data volume. Thus, a proper combination of different analytics algorithms should be defined to correctly model data under analysis. These activities are very time-consuming and require a lot of expertise to achieve the best trade-off between the quality of the result and the execution time. Innovative, scalable, and parameter-free solutions need to be devised to streamline the analytics process for large data collections.

The aim of this dissertation is to design and develop an automated data analytics engine to effectively and efficiently analyse large collections of textual data with minimal user intervention. Both parameter-free algorithms and self-assessment strategies have been proposed to suggest algorithms and specific parameter values for each step characterising the analytics pipeline. The proposed solutions have

been tailored to textual corpora characterised by variable term distributions and different document lengths. Specifically, a new engine named ESCAPE (**E**nhanced **S**elf-tuning **C**haracterisation of document collections **A**fter **P**arameter **E**valuation) has been designed and developed. ESCAPE includes two different solutions to address document clustering and topic modelling. In each proposed solution, ad-hoc self-tuning strategies have been integrated to automatically configure the specific algorithm parameters, as well as the inclusion of novel visualisation techniques and quality metrics to analyse the performances of the methodologies and help the domain experts to easily interpret the discovered knowledge. Specifically, ESCAPE exploits a data reduction phase computed through the Latent Semantic Analysis, before the exploitation of the partitional K-Means algorithm (named joint-approach) and the probabilistic Latent Dirichlet Allocation (named probabilistic approach). The former is based on dimensionality reduction of the document-term matrix representing each corpus, while the latter is based on learning a generative model of term distributions over topics. Both the joint-approach and the probabilistic model permit to find a lower dimensional representation for a set of documents with respect to the simple document by term matrix. Furthermore, ESCAPE includes several weighting strategies, which are able to measure term relevance in the same dataset by exploiting a local weighting schema (e.g. TF, LogTF) together with a global weighting schema (e.g. Entropy, IDF). Moreover, the outputs of the two methodologies are disjoint groups of documents with similar contents. To compare the results, different visualisation techniques have been integrated in ESCAPE to help the analyst in the interpretation of the ESCAPE results. The proposed engine has been tested through different real textual datasets characterised by a variable document length and a different lexical richness.

ESCAPE correctly identifies a good partition of a given corpus based on its main content, grouping the documents into well separated topics. Both the exploratory methodologies are able to split the corpora into well separated groups, both in terms of quality indices and easily-interpretable graphical representations. Through the joint-approach, based on a dimensionality algebraic phase before the application of the partitional K-Means algorithms, ESCAPE finds homogeneous partitions in terms of documents characterising each topic. In other words, this approach creates balanced clusters. Moreover, changing the weighting strategy, the end-user is able to partition the same dataset at different granularity levels. Specifically, the local weighting schema LogTF tends to find a small number of clusters. While, the local

weighting schema TF is able to characterise the corpus by identifying also the hidden subtopics of interest. Moreover, the weighting schema TF-IDF is able to create more clusters characterising sub-topics related to the major category. On the other hand, the global weighting schema Entropy is able to find less clusters but with a larger cardinality, finding only the main relevant topic associated with each partition.

On the other side, the probabilistic model tends to find more heterogeneous clusters than the joint-approach. The probabilistic approach, exploiting the semantic similarity among the produced topics turned out to outperform the current used approach to find proper numbers of clusters. Indeed, ESCAPE is able to capture the effective cohesion level of the clusters, and then properly identify the optimal number of topics. The clusters found for all the corpora are well separated, especially for certain weighting schemas such as TF-IDF. However, with respect to the joint-approach, some weighting schemas lead to very poor results, such as the Entropy-based schemas.

Possible future extensions concern the integration of other (i) algebraic data reduction algorithms, (ii) probabilistic topic modelling methods, and (iii) visualisation techniques. Furthermore, we are planning to introduce a semantic component able to support the analyst during the pre-processing phase (to reduce semantically correlated terms) and the post-processing phase (to help the analyst during the exploration of the results).

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Extracting value from data is a growing concern and many data mining and machine learning techniques have been developed to enable users to extract underlying structures and patterns, as well as to make predictions from large datasets. Data-driven knowledge discovery often requires users to interact with the system by tackling a variety of technical issues, such as selecting the best technique for the task at hand, determining optimal parameter settings and finding the best trade-off in terms of performance and computational costs. To make data analysis tool available to a broader range of end-users and help data scientists in making data analysis more effective and efficient, a new generation of data mining systems is needed.

Nowadays large volumes of heterogeneous data are continuously collected at an ever-increasing rate in various modern applications, ranging from social networks (e.g. Twitter, Facebook), digital libraries (e.g. Wikipedia), smart city environments, Internet of Things (IoT) services and so on. In addition, we are in an age of data-intensive science and we are witnessing the unprecedented generation and sharing of large scientific data sets. Indeed, the pace of data generation has now far exceeded the pace of data analysis [1]. The analysis of these data collections is challenging; indeed, this is a multi-step process in which the data scientists have the complex task of configuring the analytic system to transform data into actionable knowledge. In [2], the authors say that the most important challenges when working with large volumes of textual collections include the lack of textual corpora structure, the several pre-processing steps, and the ability to scale. A plethora of algorithms are currently available for performing a given data analysis phase, but for each one the

specific parameters have to be manually set and validated by a domain expert. Real datasets are also characterised by an inherent sparseness and variable distributions, and their complexity increases with the data volume. Thus, a proper combination of different analytics algorithms should be defined to correctly model data under analysis. This activity takes a lot of time and requires a lot of experience to get the best compromise between the result quality and the execution time. To this aim, innovative, scalable, and parameter-free solutions need to be devised to streamline the analytics process for large data collections.

The main research goal of this dissertation is to design and develop an engine that, given a textual corpus, yields groups of interesting topics together with their characterisation while masking the underlying complexity of the data analytics tasking from the end user. The end user should be free to concentrate on their core business without having to deal with the technical details about how such knowledge is actually obtained. To model data distribution and to identify interesting information-rich subsets within noisy datasets, state-of-the-art criteria have be studied and some new strategies proposed. Interesting and latent topics in a given corpus are automatically discovered and properly presented to the end-user. With the final goal of reducing the computational burden, distributed approaches have been exploited. The proposed engine addresses all the steps of the analytics pipeline properly enriched with self-tuning and self-assessment strategies. Specifically, (i) *Data characterisation and preparation*, (ii) *Automated algorithm configuration*, and (iii) *Knowledge navigation and exploitation* are included in the engine by enhancing state-of-the-art algorithms with strategies that automatically take care of specific parameter setting and knowledge quality evaluation.

**Data characterisation and preparation.** To characterise data distribution, a set of descriptive information has been defined. Specifically, innovative criteria are developed to model data distributions by exploiting statistical indexes and underlying data structures to highlight hidden data knowledge. Moreover, once the corpora are collected, they have to be properly pre-elaborated. Pre-processing is an important and critical task that affects the quality of the text mining results. It includes different steps that are performed sequentially as interrelated tasks. These steps are needed for transferring text from human language to machine-readable format for further processing.

**Automated algorithm configuration.** In the literature, numeral alternative algorithms are available for performing a given data mining task, and in most cases no algorithm is universally superior. Various aspects influence algorithm performances, such as input data cardinality, its distribution, and the kind of extracted knowledge (i.e., type of analysis to be performed). To automatically lead the analyst's exploration of the search space to a different level (e.g. parameter setting), a set of specific metrics has been studied to evaluate and compare the goodness of knowledge discovered by different algorithm runs. Furthermore, when dealing with large data collections, the computational cost of the data mining process (and in some cases the feasibility of the process itself) can potentially become a critical bottleneck in data analysis.

**Knowledge navigation and exploitation.** A data mining process performed on large databases may lead to the discovery of a huge amount of knowledge, which is usually hard to process and analyse. Nevertheless, an in-depth analysis may be required to identify only the most actionable knowledge. The characterisation of the significance of knowledge in terms of unconventional statistical criteria should be also address to evaluate and compare different solutions. Furthermore, ad-hoc visualisation approaches should help the domain expert in understanding the data under analysis along with the extracted knowledge and inference. In addition, informative dashboards allow users to interpret and explore data content, identify interesting hidden patterns, infer causation and correlation, and support activities that are not always possible with traditional data analysis techniques.

The main focus of this dissertation is on the analysis of textual data collections, including the main obtained achievements. To this aim, I have designed and developed ESCAPE (**E**nhanced **S**elf-tuning **C**haracterisation of document collections **A**fter **P**arameter **E**valuation), a distributed self-tuning engine running on Apache Spark, able to cluster a collection of textual documents into cohesive and well-separated groups with minimal user intervention. It includes all the analytics blocks to make the overall analysis problem more effectively tractable, including innovative strategies to relieve the end-user of the burden of selecting proper values for algorithm-specific parameters.

## 1.1 Dissertation plan and research contribution

The main achievement obtained from my PhD activities is threefold.

(i) First, I have designed and developed ESCAPE (Enhanced Self-tuning Characterisation of document collections After Parameter Evaluation), a distributed self-tuning engine running on Apache Spark, able to cluster a collection of documents into cohesive and well-separated groups with minimal user intervention. The preliminary version of the proposed methodology has been presented in [3]. It includes all the analytics blocks to make the overall analysis problem more effectively tractable including innovative strategies to relieve the end-user of the burden of selecting proper values for algorithm-specific parameters [4, 5]. Up to now, this research activity led to publish 5 papers included in international conference proceedings published by IEEE [4, 6], ACM [3, 5], and in CEUR Proceedings [7]. I have presented papers [5] and [6].

(ii) I have designed and developed METATECH (METeorological data Analysis for Thermal Energy CHaracterisation), a data mining engine including different data analytics algorithms, devised to build transparent and self-tuning models correlating energy-related data. This research activity led to publish a paper published in an international journal [8]. Other research activities focused on the analysis of energy-related data have been included in international conference proceedings published by IEEE [9] and in CEUR Proceedings [10]. I have presented both papers.

(iii) I have participated as data scientist to several research projects funded by international private companies, fostering national and international collaborations. Some research results have been published on papers included in international conference proceedings published by IEEE [11, 12], by Springer [13] and in CEUR Proceedings [14]. I have presented paper [11] at the first International Conference on Smart Energy System and Technologies which won the best paper award.

### 1.1.1 Dissertation plan

This dissertation is organised as follows. First, a literature review of basic text mining concepts and methods is provided in Chapter 2. Chapter 3 discusses the proposed engine, named ESCAPE with its main building blocks. Chapter 4 reports

the experimental evaluation performed to assess the engine. Finally, Chapter 5 draws the final conclusions and presents future developments of this research activity.

# Chapter 2

# State-of-the-art

In this Chapter, a detailed analysis of the main methodologies used to analyse unstructured data with exploratory approaches is reported. The two main activities when an analyst deal with textual corpora are (i) *clustering methods* and (ii) *topic modelling methods*.

*Clustering methods* aim at partitioning data into coherent groups. A key aspects for clustering is the definition of a distance for the data instances. Similarity or distance among two textual data documents is usually measured according to a notion of similarity/distance in the space describing the document terms. The cosine similarity is the most well-known similarity measure exploited in textual doamin, as discussed in [15]. However, the Euclidean distance can also be exploited after normalising the document vectors. Thus, the Euclidean distance is usually used to measure the distance among documents. An interesting approach is to reduce the dimensionality of the dataset under analysis, though the application of algebraic models. We will define this approach, *joint approach*, since we apply a data reduction algorithm before the application of a clustering algorithm. On the other hand, the purpose of *topic modelling methods* is to discover the latent themes (topics) assumed to have generated the documents of a corpus.

*Topic modelling* methods are built on the distributional hypothesis, suggesting that similar words occur in similar contexts. Section 2.3.1 describes algebraic models to improve the clustering analysis, while Section 2.3.2 refers to probabilistic models. Lastly, Section 2.4 reports the main visualisation techniques used to represent textual data content.

## 2.1 5 V's of Big Data

The term Big Data seems to be popping up everywhere in the last years. However, *big data* is often used to refer to any dataset that is difficult to manage using traditional approach (e.g. traditional database systems) or for any collection of data that is too large to process on a single server. Of course, the specification of the term *big* is elusive. As a matter of fact, what is considered big for one organisation may be small for another. Moreover, the complexity of the data has a very important role that must be considered.

Data scientists almost describe big data [16] as having at least three distinct dimensions: (1) volume, (2) velocity, and (3) variety. Nowadays, more Vs have been added to the list, to also include (4) variability and (5) value.

1. **Volume.** The name Big Data itself is related to a size which could be enormous [17]. Volume represents a huge amount of data. To determine the value of such kind of data, the size plays a very crucial role. If the volume of data is too large then it is actually considered as a Big Data. This means whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data. In particular, text corpora are a very important example of this issue, since nowadays large volume of textual data are continuously collected in several domains.

2. **Velocity.** It refers to the high speed of accumulation of data [18]. In Big Data velocity data flows in from sources like machines, networks, social media, mobile phones and so on. There is a massive and continuous flow of data. This determines the potential of data based on how fast the data is generated and processed to meet the demands. Of course, in the text mining domain, the velocity is a crucial aspect, since social networks and digital libraries are growth.

3. **Variety.** It refers to the possible heterogeneous nature of data which is (i) structured, (ii) semi-structured and (iii) unstructured data [19]. Variety is basically the range of data types and source.

    - *Structured data*: This data is basically an organised type of data. It generally refers to data that has defined the length and format of data.

- *Semi-Structured data*: This data is basically a semi-organised data. It is generally a form of data that do not conform to the formal structure of data. An example of this type of data are the Log files.
- *Unstructured data*: This data basically refers to unorganised data. It generally refers to data that does not fit neatly into the traditional row and column structure of the relational database. Texts, pictures, videos are the examples of unstructured data which cannot be stored in the form of rows and columns.

The unstructured data are one of the most complex types of data to be analysed. This dissertation focuses on the automatic analysis of this data, including self-tuning strategies able to help the non-expert users to be able to analyse their collections, without knowing the complexity behind the algorithms.

4. **Variability.** It refers to inconsistencies and uncertainty in data, that is data which is available can sometimes get messy and quality and accuracy are difficult to control. Big Data is also variable because of the multitude of data dimensions resulting from multiple disparate data types and sources. In particular, the management of textual data is complex also due to the variability of the information contained within it. Furthermore, the vocabularies that describe large collections of documents are enormous, increasing the size of the data.

5. **Value.** The bulk of data having no value is of no good to the company, unless you turn it into something useful [20]. Data in itself is of no use or importance but it needs to be converted into something valuable to extract Information [21]. It is necessary to bring hidden knowledge to the surface and transform the data into actionable knowledge [22].

The above V's are the dimensions that characterise big data, and also embody its challenges: we have huge amounts of data, in different formats and varying quality, that must be processed quickly. Moreover, data is being generated all the time at ever faster rates. The analytics pipeline for the analysis of textual data is not unique. To this aim, this dissertation proposes a fully automatic approach to designing text analysis pipelines for arbitrary information needs that are optimal in terms of run-time efficiency and that robustly mine relevant information from text of any kind.

It is important to note that the goal of processing big data is to gain insight to support decision-making. It is not sufficient to just be able to capture and store the data. The point of collecting and processing volumes of complex data is to understand trends, uncover hidden patterns, detect anomalies, and so on. By this way, the analysts have a better understanding of the problem being analysed and can make more informed.

To address the challenges of big data, innovative technologies are needed [23]. Parallel, distributed computing paradigms, scalable machine learning algorithms, and real-time querying are key to analysis of big data. Distributed file systems, computing clusters, cloud computing, and data stores supporting data variety and agility are also necessary to provide the infrastructure for processing of big data.

With all the data generated from social media, smart sensors, satellites, surveillance cameras, the Internet, and countless other devices, big data is all around us. The endeavor to make sense out of that data brings about exciting opportunities indeed [24].

## 2.2 Text mining

*Text Mining* could be defined as the process of extracting knowledge from unstructured data to gather valuable information. Text mining, also known as text data mining, involves the use of several data mining algorithms, machine learning (ML), statistics, and natural language processing (NLP), with the final aim of extracting high quality, useful and hidden information from the data. Indeed, in these masses of continuously increasing information, which is poorly catalogued and organised (or not organised at all) is almost impossible to navigate through them to search or draw any interesting conclusion or meaning. This problem is also known as *info-glut problem* [25].

In the recent year, a tremendous and huge increase in adoption of text mining for business applications has been seen. The main reason being by increasing awareness about text mining which is available today. Text analytic can help analytics businesses by extracting insights from free textual texts written by (or about) customers, combining it with feedback data (if available), and identifying patterns and trends. Nowadays, due to the enormous volume and the complexity of the available data, manual analysis alone is not able to capture this level of insight.

In text mining, the analyst should analyse the data to capture key concepts and themes and uncover hidden relationships without a-priori knowledge of the precise words (i.e., terms) that the authors have used to express those concepts. There are actually two complex issue when we talk about text mining (i) *synonymy* and (ii) *polysemy*. The vector space model of documents does not address the issue of synonymy or polysemy that is an inherent part of documents. Synonymy is when there are many ways to refer to the same object using different words. People could use several terms to describe the same information in different contexts (e.g. the following list of words: *schoolboy*, *lad*, and *stripling* belong to the concept of *boy*). On the other hand, polysemy refers to terms that have multiple meanings (i.e., homography), not related to each other (*Java* can refer to a computer programming language, but it also can refer to *coffee*). Several algorithms have been proposed to deal with these kinds of problems.

Text mining, coupled with data mining algorithms, offers better insights than adopting any one of the two [26]. However, a right understanding of both, before combining data mining with text mining should be presented. This process typically includes the following steps:

First, the analyst should identify the text corpora to be exploited. Next, the text should be mined and transformed into a structure data. The text mining algorithms should be now applied to the source text. It is necessary to build concept and category models for the data that is mined, identifying the key concepts and create separate categories for each of them. The number of concepts from the unstructured data could be too many in number. In this case, it is advisable to identify the most popular or talked about concepts (i.e., the hot topics [27]). Finally, the use of standard data mining techniques, such as clustering, classification, and predictive modelling, could be useful to discover relationships between the concepts.

There is no a single pipeline to analyse textual data. Moreover, several strategies have been proposed in the last decades to build model able to divide textual collections into homogeneous groups of documents related to similar topics. However, no text analysis tool implements all levels, nor any processing workflows [2]. This plethora of algorithms needs a lot of expertise to be configured. Furthermore, the problem of automatically analyse and set right parameters for the entire text mining process is a very complex task. Several authors have proposed different solution for each algorithm.

Large volumes of textual data are being collected at an ever increasing-rate in various modern applications (e.g. social networks like Twitter, Facebook, e-learning platforms, digital libraries) [5]. However, their exploitation is limited without effective approaches able to automatically discover useful information from textual data collections with limited user intervention. The text mining is focused on studying algorithms to find implicit, previously unknown, and potentially high-quality information from a large collection of documents. Text mining activities include: (i) grouping documents with similar properties or similar content [5, 28], (ii) topic modelling [4, 29] and detection [30], (iii) classification models [31, 32], (iv) opinion mining and sentiment analysis [33], (v) document summarisations [34, 35], and (vi) document querying [36].

Applying the above techniques to large data collections has often entailed coping with a critical bottleneck of computational costs. To address this issue, many research efforts have been devoted to designing innovative algorithms and methodologies to support large scale analytics based on MapReduce [37]. A step further towards a most promising analytics framework is Apache Spark [38] that outperforms Hadoop performance thanks to its distributed memory abstraction, a key aspect for data analytics algorithms. Applications of these techniques to text mining become natural, given the large volume of textual data generated every day by a large variety of applications.

In the scientific research, several approaches and solutions have been attempted and proposed in order to firstly represent, and then mine and retrieve information [39] from the text sources. Depending on the modelling of the text data and the used techniques, different models have been proposed in the scientific literature: set-theoretic [40] (such as the Boolean models, representing documents as sets of words or phrases), algebraic [5, 3, 41] (representing documents as vectors or matrices, such as the Vector Space models and the Latent Semantic Analysis) and probabilistic [42, 43] (such as the Latent Dirichlet Allocation, representing documents as probabilities of words). However, besides the approach used to analyse the text documents, given the huge amount of data available, making the mining activity automatic is the natural subsequent step in information retrieval. Indeed, exploiting the data mining techniques to extract the hidden information in the collected data is not effective if a constant human supervision of the activity is needed. This happens also in the case of text mining. Indeed, being text mining a multi-step process requiring specific configurations and parameters for each algorithm involved in the analysis,

the presence of texts-field expertise and analysts should be required to guide the retrieving process. To overcome this problem, innovative solutions are needed to make the analysis of large data scalable and not supervised by human analysts and data experts more effectively treatable.

### 2.2.1 Text mining applications

Text mining is nowadays used to answer interesting, business questions and to optimise day-to-day operational efficiency; but also to improve long-term strategic decisions in the automotive, health and financial sectors. Methodologies like categorisation [44], entity extraction [45], and sentiment analysis [46] are usually used to identify insights, patterns, and trends in large volumes of unstructured data. Below, a few real-life examples of text mining have been discussed.

**Risk management.** An inadequate risk analysis takes into account the main reasons for failure in any industrial sector. However, text mining could be a great ally, helping us to solve the challenging problem of a robust risk analysis. Especially in the financial sector, risk management software based on text-mining technology can greatly increase the risk mitigation capacity [47] that ensures complete management of large databases and link information. Furthermore, these technologies may be able to access the right information at the right time.

**Knowledge Management.** Managing large volumes of data makes it difficult to find specific information. A classic example of this important problem is found in the health sector [48]. Here, professionals have at their disposal a great deal of information (e.g. years of research in genomic or molecular techniques, volumes of clinical data on patients) that could potentially be used for the development of new products. Knowledge management software based on text mining could offer a clear and reliable solution to the problem of information influence, thanks also to advanced search and querying algorithms.

**Prevention of Cybercrime.** The cybercrime burden [49] is often supported by the random availability of data on the Internet and consequential exchanges. The unidentified criminal soon becomes untraceable. Thanks to text mining, intelligence and anti-crime applications keep cybercrime at bay. Companies and law enforcement or intelligence agencies use text-mining techniques to analyse the origin and nature of data extraction.

**Customer Care Service.** Text mining and natural language processing are widely used for customer care applications [50], [51]. In fact, the adoption of text analysis techniques guarantees a better customer experience using various sources of valuable information such as surveys, problem tickets and customer call notes to optimise the quality, effectiveness and speed in solving problems. Moreover, to improve the automatic customer response, it is necessary to carry out the analysis to drastically reduce the dependence on the operations of the call center, thus being a valid support for the analyst.

**Spam Filtering.** E-mail is an effective, fast and reasonably cheap way to communicate, but it presents a dark side: spam. Today, spam e-mails are a sensitive area for most Internet service providers [52], taking into account the higher cost of service management and software/hardware updating. For each user, spam is an entry point for viruses and impact productivity. Text extraction algorithms are implemented to improve the effectiveness of filtering methods based on statistical features [53].

**Social Media Data Analysis.** Social media (e.g. Twitter, Facebook), which represent the most prolific source of unstructured data, are considered as a valuable source of market and customer intelligence. Several companies are using text mining techniques to analyse or predict customer needs and understand the perception of their brand [54]. In this way, in this large volume of data, we can extrapolate opinions, feelings and emotions and people relationships with products and brands.

**Business Intelligence.** To uphold and support the decision making, the company could be interested in applying text analytics tasks. Text mining helps in faster and better analysis. By this way, only the relevant content can be extracted from the large data volume, helping the analysts to take the best marketing decisions [55].

**Contextual Advertising.** Digital advertising is a moderately new and growing application field for text analytics [56]. Indeed, companies have made text mining as the main and core engine for contextual re-targeting with great success and results. Moreover, compared to the traditional cookie-based approach, contextual advertising offers better accuracy and total security, completely preserves the user's privacy.

The increasing scope of the web and the large amount of electronic data piling up throughout the web has provoked the exploration of hidden information from their text content[57]. News articles published on different news portals throughout the web are the sources of the information. These can also be very good topics for the research on text mining. Clustering of similar news headlines and putting them under

a single platform with the corresponding links to the news portal sites can be a very efficient option to the exploration of the same news article across multiple different news portals, which is, in fact, a tedious and time-consuming task.

The study of the clustering problem precedes its applicability to the text domain. Traditional methods for clustering have generally focused on the case of quantitative data, in which the attributes of the data are numeric. The problem has also been studied for the case of categorical data, in which the attributes may take on nominal values [58].

The problem of clustering and topic modelling finds applicability for a number of tasks:

1. **Document Organization and Browsing:** The hierarchical organization of documents into coherent categories can be very useful for systematic browsing of the document collection. A classical example of this is the Scatter/Gather method [59], which provides a systematic browsing technique with the use of clustered organization of the document collection.

2. **Corpus Summarisation:** Clustering techniques provide a coherent summary of the collection in the form of cluster-digests [60] or word-clusters [61, 62], which can be used in order to provide summary insights into the overall content of the underlying corpus. Variants of such methods, especially sentence clustering, can also be used for document summarisation, a topic, discussed in detail in Chapter 3. The problem of clustering is also closely related to that of dimensionality reduction and topic modeling.

3. **Document Classification:** While clustering is inherently an unsupervised learning method, it can be leveraged in order to improve the quality of the results in its supervised variant. In particular, word-clusters [61, 62] and co-training methods [63] can be used in order to improve the classification accuracy of supervised applications with the use of clustering techniques.

4. **Recommender system:** is an information filtering system that seeks to predict the rating or preference that a user would give to an item [64, 65]. They are primarily used in commercial applications; now are utilised in a variety of areas and are most commonly recognised as playlist generators for video and music services like Netflix, YouTube and Spotify, product recommenders for

services such as Amazon, or content recommenders for social media platforms such as Facebook and Twitter [66]. These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries. There are also popular recommender systems for specific topics like restaurants and online dating.

## 2.3 Document clustering and topic modelling

In the following Subsections, we will review some of the theory of the main algorithms used in literature to divide collection of textual corpora into groups of documents related to specific topics, the main drawbacks of each methodology are also discussed. We divide them into two main categories: (i) *joint-approach* (including an algebraic model for the reduction and an unsupervised technique for the clustering phase) and (ii) *probabilistic models*. As described before, text data are by their nature very variable and dirty, differing a lot based on the typology, the source, the target and the field of expertise. Because of this, before analysing the documents several pre-processing steps are needed. These steps are described in the Chapter 3. The most relevant preprocessing steps, applied automatically, are the following: (1) *Document splitting* divides each document into sections or paragraphs according to the analytics task; (2) *tokenisation*breaks text into discrete words within the same sentence; (3) *stopwords removal* eliminates non meaningful words (e.g. articles, prepositions, and conjunctions) that frequently occur in the text but are not informative words; (4) *stemming* removes prefixes and suffixes to normalise words to their base or root form (stem or term). After the pre-processing, the documents are represented in the *bag of words* form [67], that describes texts disregarding the terms order and the grammar rules, but however representing the main themes. In order to better identify the correct topic of a document and help the clustering process to group similar documents together, weights can be assigned to all the terms in the corpus. The weights measure the relevance the terms have in the documents, and they are computed as the product of a local and a global weight. The weight that a word has within the document is called local weight, while the weight it has with respect to the whole corpus is called global weight. A weighting function applied on a collection $D$ generates its weighted matrix $X$. Specifically, for each term $t_j$ of a document $d_i$ the corresponding weight $x_{ij}$ in $X$ is computed as the

product of a local term weight ($l_{ij}$) and a global term weight ($g_j$) ($x_{ij}$=$l_{ij} * g_j$). In ESCAPE we have integrated three local term weights: *Term-Frequency* (TF) [68], *Logarithmic term frequency* (Log) [69] and *Binary* (Boolean) [70] and three global term weights: *Inverse Document Frequency* (IDF) [68], *Entropy* (Entropy) [69] and *Term-Frequency* ($TF_{glob}$) [70].

### 2.3.1 Joint-approach

In the joint approach, two unsupervised approaches are applied to the matrix created at the previous step. First, a reduction algorithm (e.g. Singular Value Decomposition [71], Principal Component Analysis [72]) is applied to the collection to construct a low-rank approximation of the original matrix. Then, a clustering algorithm is applied to make the problem more effectively tractable. In order to reduce the dimensionality of the corpus and focus the computation only on the most relevant concepts of the documents, a data transformation is needed.

#### Singular Value Decomposition

The main reduction algorithm used in literature is the Singular Value Decomposition [71]. When this technique is applied to the document-term matrix in the textual contest, is also known as LSA (Latent Semantic analysis) [73]. LSA allows reducing the dimensionality of matrix $X$ (i.e., the document-term matrix) while disregarding some irrelevant dimensions [74]. The choice of the correct dimensionality reduction, without losing significant information, is an open research issue and a very complex task [5, 70]. LSA is able to analyse relationships between groups of documents and terms generating sets of concepts in the corpus under analysis. Through the application of the Singular Value Decomposition (SVD), the analyst finds the hidden concepts. Too few dimensions after the LSA process will lead to poor data representation, whereas too many dimensions will result in more noisy data. LSA has been introduced to face the problem of how to find relevant documents from search words. The fundamental difficulty arises when words are compared to find relevant documents, because what should be compared is the meanings or concepts behind the words. LSA attempts to solve this problem by mapping both words and documents into a concept-space and the comparison is done in this new space [75]. In order to make this problem more effective tractable, some simplifications are introduced:

- Documents are represented as *bags of words*, where the order in which the words appear in the documents is not important. Only the frequency is relevant to measure the weight of terms in the corpus.
- Concepts are represented as patterns of words that appear together in the collection.

SVD is a matrix factorisation method that decomposes the original matrix (document-term matrix) $X$ into three matrices ($U$; $S$; $V^T$). $U$ is a $d \times r$ column-orthonormal matrix (i.e., $U^T U = I$), $S$ is a d×d diagonal matrix and $V$ is a $r \times t$ column-orthonormal matrix (i.e., $V^T V = I$). $S$ is also called the concept-matrix, while $U$ and $V$ are called document-concept similarity matrix and term-concept similarity matrix, respectively. Each cell of the weighted matrix $X$ is represented as $x_{ij} = \sum_{c=1}^{r} d_{i,c} \lambda_c t_{c,j}$, where each weighted term $t_i$ in document $d_j$ is expressed as a linear combination of term-concept and document-concept weights. We obtain the exact decomposition (lossless representation) of the original matrix in Equation 2.1.

$$X = USV^T \tag{2.1}$$

The matrix $S$ includes a singular value for each dimension (term) in the document collection under analysis. The significance of each dimension is represented by the magnitude of the corresponding singular value in $S$. Through the SVD decomposition some insignificant dimensions in the transformed space can be easily identified to approximate (in the least square sense) matrix $X$. Insignificant dimensions in $S$, which are expressed by a low magnitude of singular values, may represent noise in the data and should be disregarded in the subsequent analysis steps. The singular values model the relative importance of the dimensions. As the singular values decrease, so does the effect of the dimension. The $k$ selected singular values correspond to the hidden concepts.

Since both $U$ and $V$ are orthonormal, we can multiply both the side of Equation 2.1 by $V$, we obtain $XV = US$. It can be seen as a projection of documents in the r-dimensional concept space, where $r$ is the rank of the original matrix $X$. In this new space, documents are represented by the row of $US$. Given a target dimensionality $k$ (normally $k << r$), it is possible to obtain an optimal approximation of the original matrix by retaining only the *top-k* largest singular values of the matrix $S$. Among all the rank-k approximation, we analyse the one that minimises the Frobenius

norm. However, LSA has no theoretical optimal reduced dimension [76], and its computational estimation is difficult without the potentially expensive process of trying many test cases.

To identify the main relevant dimensions ($K_{LSA}$) in $X$, we proposed an innovative algorithm (See Chapter3). Given $K_{LSA}$, usually only the largest singular $K_{LSA}$ values in $S$ are used and the remaining ones are set to zero. The approximated matrix of $X$, denoted $X_{K_{LSA}} = U_{K_{LSA}} S_{K_{LSA}} V^T_{K_{LSA}}$ is obtained by reducing all three decomposed matrices $(U, S, V^T)$ to rank $K_{LSA}$. In general, the low-rank approximation of X by $X_{K_{LSA}}$ can be viewed as a constrained optimisation problem with respect to the constraint that $X_{K_{LSA}}$ have rank at most $K_{LSA}$. When forced to squeeze the terms-documents down to a k-dimensional space, the SVD should bring together terms with similar co-occurrences. This intuition suggests that the dimensionality reduction could improve the results [5, 77]. The graphical representation of the reduction is reported in Figure 2.1. The original document-term matrix $X$ is decomposed into three distinct matrices. In our hypothesis, the number of documents is less than the number of distinct terms in each collection.

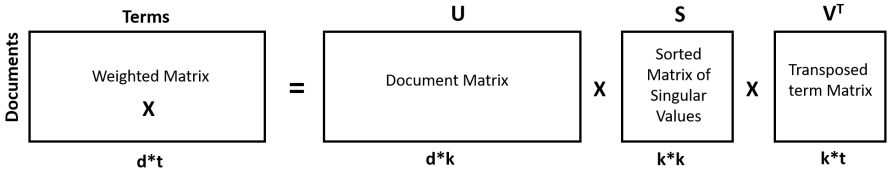



Fig. 2.1 SVD decomposition.

### Principal Component Analysis

Principal Components Analysis (PCA) [72] is a well-known statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables, into a few linearly uncorrelated variables called principal components. This mathematical transformation was invented in 1901 by Karl Pearson [72] and expanded by Hotelling [78]. The discussion of the previous technique (i.e., SVD) could remind similar aspects of PCA. As the SVD decomposition, PCA is useful for arbitrary rectangular matrices such as the document-term matrix $X$.

From the rectangular matrix $X$, PCA specifies that the square covariance or correlation matrix[1] $C$ is formed and then the eigenvalue decomposition of $C$ is computed through the following decomposition:

$$C = X * X^T = D * \Delta * D^{-1}$$

where $\Delta$ represents the eigenvalues and $D$ the eigenvectors necessary to project the original documents into the reduced space. These eigenvectors are also known as *principal components*, while their associated eigenvalues are called *principal values*. Moreover, the $i^{th}$ principal value is proportional to the additional variance described by adding the $i^{th}$ principal component. By this way, the ratio

$$v_i = \frac{\lambda_i}{\sum_{j=1}^{r} \lambda_j}$$

indicates the amount of variation captured and described by the $i^{th}$ component.

To analyse the impact of each eigenvalue (i.e., the amount of variance until the $i^{th}$ component), the *scree plot* of the principal component can be plot [79]. The scree plot is usually used to analyse and visualise the amount of the variance explained by using only $k$ principal components. As for the SVD, the number of components to be chosen is a complex task [3].

Although based on similar procedures, PCA and SVD approaches operate on different data, and they do not produce the same results. Depending on whether the raw data (i.e., the document-term matrix $X$) or the covariance matrix is used (i.e., the covariance matrix of $X$), the basis vectors found for the reduced space could be different, respectively. Furthermore, the square roots of the eigenvalues of $X * X^T$ are the singular values of $X$.

In [79], a very interesting example is reported to analyse how the principal components change from one approach to the other one. In Figure 2.2 is reported this example, where a trivial dataset is used. We consider only documents formed by only two words (i.e., Word A and Word B). In this way, we are able to plot both the original and the reduced spaces to see the difference between them. If the PCA com-

[1] We recall that the covariance matrix of $X$ can be computed by subtracting the mean of each column of the matrix from each cell in that column to form the new matrix which is less a scale factor. The correlation matrix is typically used when the variables represent measurements based on differing scales. This is not the case for the document-term frequency matrix.

ponent maximises the variance, the SVD finds the best fitting line in the least-squares sense. In Figure 2.2 the first and the second principal components and singular vectors are drawn together [79]. However, in both procedures, the second component is always orthogonal to the first one. It can be seen that the two components are able to capture the entire space, and the projection of each document places it in the same position. The only main difference is that now are used different coordinate systems (either the two SVD or the two PCA lines serve as the new axes) to represent the document positions.

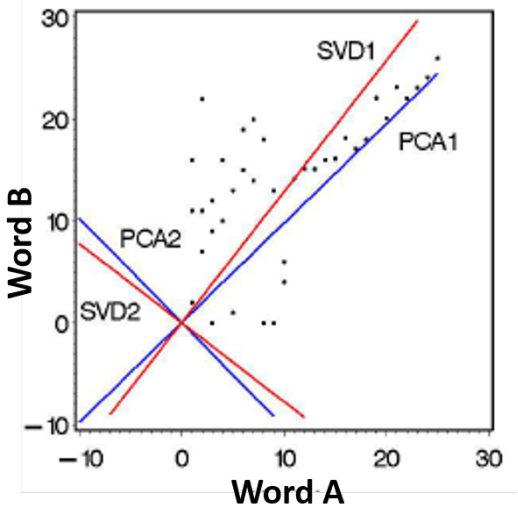



Fig. 2.2 First and Second Principal Components for the SVD and PCA approaches [79].

In addition, the variance criterion of the principal values holds in relation to the SVD. As a matter of fact, the amount of additional variance explained by adding the $i^{th}$ singular vector is quantified by the square of each singular value. However, the possible decision of using $k$ dimensions based on the proportion $p_k = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2}$ is not as applicable for the SVD (as it is for PCA) because the criteria for selecting each additional component using the singular value decomposition technique is not based on maximising the explained variance as it is done for the principal components.

### Sparse latent analysis

A great research effort has been done in [80–82] to analyse textual corpora using sparse latent analysis. Specifically, in [80], the authors focus on the use of sparse

PCA [81] and Elastic Net regression for extracting intelligible topics from a big textual corpus and for obtaining time-based signals quantifying the strength of each topic in time. Sparse machine learning has recently emerged as powerful tool to obtain models of high-dimensional data with high degree of interpretability, at low computational cost [82]. The approach has been successfully used in many areas, such as signal and image processing.

### Autoencoder

There are several ways to reduce the dimensions of large data sets to ensure computational efficiency such as backwards selection, removing variables exhibiting high correlation, high number of missing values, principal components analysis, singular value decomposition and so on.

A relatively new method of dimensionality reduction is the autoencoder [83]. Autoencoders are a branch of neural network which attempt to compress the information of the input variables into a reduced dimensional space and then recreate the input data set. Typically the autoencoder is trained over number of iterations using gradient descent, minimising the mean squared error. The key component is the bottleneck hidden layer. This is where the information from the input has been compressed. By extracting this layer from the model, each node can now be treated as a variable in the same way each chosen principal component is used as a variable in following models. Auto Encoders are is a type of artificial neural network used to learn efficient data patterns in an unsupervised manner [84]. An Auto Encoder ideally consists of an encoder and decoder. The Neural Network is designed compress data using the Encoding level. The Decoder will try to uncompress the data to the original dimension. To achieve this, the Neural net is trained using the Training data as the training features as well as target. In [85] the authors present a training method that encodes each word into a different vector in semantic space and its relation to low entropy coding. Elman network [86] is employed in the method to process word sequences from literary works. The trained codes possess reduced entropy and are used in ranking, indexing, and categorising literary works. A modification of the method to train the multi-vector for each polysemous word is also presented where each vector represents a different meaning of its word. These multiple vectors can accommodate several different meanings of their word.

### Clustering Algorithms

The different document-concept vectors could also be clustered using a clustering algorithm such as K-Means. The difference between clustering and LSA is that clustering algorithms assign each document to a specific cluster, while LSA assigns a set of topic loadings to each document. However, a clustering algorithm applied after the singular value decomposition improve the results, as shown in [3, 5]. As a matter of fact, the large dimensions of data become obstacles. So, the singular value decomposition is applied to data to reduce the dimension of the data prior to the learning process, using the clustering phase.

Clustering or cluster analysis is a multivariate analysis technique which goal is grouping objects so that objects in the same group (in the same cluster) are more similar to each other (using some distance) than object assigned to different groups. In cluster analysis, there is no prior information about the group or cluster membership for any of the objects. In many real case, however, clustering is only a useful starting point for other interesting purposes, such as classification, data summarisation, and so on. Indeed, it is an exploratory data mining algorithm but also a common procedure for statistical analysis, however today it is used in many other emerging fields, including pattern recognition [87], machine learning [88, 89], computer graphics [90], information retrieval [91], biology [92, 93].

A simple notion of a cluster cannot be precisely defined, which is one of the reasons why there are several clustering algorithms [94]. The common denominator between all the algorithms is the final aim of grouping data objects into well separated clusters. However, different researchers employ different cluster models, and for each of these cluster models again different algorithms can be given. The notion of a cluster, as found by different algorithms, varies significantly in its properties. Understanding these cluster models is key to understanding the differences between the various algorithms. Typical cluster models include:

- **Partitional Algorithms.** A partitional clustering [95, 96] simply divide the set of data objects into non-overlapping subsets (clusters) so that each data object falls in exactly one cluster.
- **Hierarchical Algorithms.** A hierarchical clustering [97, 98] produces a set of nested clusters which are organised as a hierarchical tree. Strategies for

hierarchical clustering generally are divided into two types: (i) agglomerative and (ii) divisive.

- **Density-based Algorithms.** In density-based clustering [99, 100], clusters are defined as areas of higher density than the remainder of the dataset. Data objects which belong to these sparse areas are usually considered to be noise and border points [101].

- **Graph-based.** In Graph-based clustering algotithms [102] data objects are represented as nodes in a complete or connected graph, also called *clique*. In the mathematical research area of graph theory, the clique is a subset of nodes such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster [103].

K-Means is one of the simplest unsupervised learning algorithms that solves the well-known clustering problem [104]. It is a simple and partitional strategy that attempts to find $K$ clusters, represented by their centroids, given by the mean value of the objects (i.e., textual documents in this dissertation) in each cluster. Initially, the partitional algorithm randomly chooses $K$ documents of the collection as centroids. Then, each document is assigned to the cluster whose centroid is the nearest to that document. Finally, the mean of all the documents in each cluster is computed to recalculate the new centroids. The process iterates until the centroids do not change. Unlike other algorithms (e.g. hierarchical clustering), K-means is computationally faster and produces tighter clusters, especially if clusters are globular. However, K-Means requires the a-priori knowledge of the number of clusters, which is usually hard to define [105]. The similarity between two documents is usually measured according to a notion of similarity/distance in the space describing the document terms. Although the cosine similarity is the most common similarity measure exploited, as discussed in [15], the Euclidean distance can also be used after normalising the document vectors with respect to the Euclidean norm. Thus, the Euclidean distance is usually exploited to measure the distance among documents. Indeed, for normalised vectors cosine similarity and Euclidean similarity are connected linearly.

Cosine distance is actually cosine similarity and it is computed as $\cos(\mathbf{x},\mathbf{y}) = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2 \sum y_i^2}}$, where $\mathbf{x}$ and $\mathbf{y}$ are two vectors. With euclidean distance for normalised

vectors we obtained (i.e., $\sum x_i^2 = \sum y_i^2 = 1$):

$$||x-y||^2 = \sum (x_i - y_i)^2 = \sum (x_i^2 + y_i^2 - 2x_i y_i) =$$

$$\sum x_i^2 + \sum y_i^2 - 2\sum x_i y_i = 1 + 1 - 2cos(x,y) = 2(1 - \cos(x,y)).$$

Note that for normalised vectors $\cos(\mathbf{x},\mathbf{y}) = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2 \sum y_i^2}} = \sum x_i y_i$. This demonstrates that there is a direct connection between these two distances for normalised vectors. After turning each vector into a unit vector, the partitional K-Means algorithm is applied.

### 2.3.2 Probabilistic model

A completely different approach from the one presented in the previous Section, is the probabilistic topics modelling approach. This technique represents textual documents as probabilities of words and aims to discover and annotate large archives of texts with thematic information. Probabilistic topic modelling algorithms are based on statistical methods that analyse the original texts and their words in order to discover the arguments they go through, and to which other documents they are related. These algorithms do not require any a-priori annotation or labelling of the documents, but they are able to describe corpora of documents without previous knowledge of the datasets. In this subsection, we analyse the two main probabilistic models used in literature: (i) **probabilistic Latent Semantic Analysis** (pLSA) and **Latent Dirichlet Allocation** (LDA). For each algorithm, a detailed description is reported, including also the main drawbacks of these methods.

#### Probabilistic Latent Semantic Analysis

*Probabilistic latent semantic analysis* (pLSA) [43], also known as probabilistic latent semantic indexing (pLSI, especially in information retrieval circles) is a statistical technique for the analysis of two-mode and co-occurrence data. In effect, one can derive a low-dimensional representation of the observed variables in terms of their affinity to certain hidden variables, just as in latent semantic analysis, from which pLSA evolved.

Compared to standard latent semantic analysis which stems from linear algebra and downsizes the occurrence tables (usually via a singular value decomposition), probabilistic latent semantic analysis is based on a mixture decomposition derived from a latent class model [106]. This results in a more principled approach which has a solid foundation in statistics.

It was developed in 1999 by Thomas Hofmann [43] and it was initially used for text applications (such as indexing, retrieval, clustering); however nowadays its application spread in other fields such as computer vision [107–109] or audio processing [110].

pLSA can be considered in two apparently different ways:

- **Latent variable model**. The probabilistic structure of pLSA is based on a statistical model, called the *aspect model*. The latent/hidden variables (which are represented by arguments/concepts) are associated to the observed variables (represented by documents and words, for the text domain).
- **Matrix factorisation**. Similar to Latent Semantic Analysis (LSA), pLSA aims to reduce the co-occurrence matrix to reduce its dimensionality. However, pLSA is usually seen as a healthier method because it provides a probabilistic interpretation, whereas LSA achieves the matrix factorisation using only mathematical bases (more precisely, LSA uses the singular value decomposition method).

The goal of the pLSA is to use the co-occurrence matrix to extract the topics and explain the documents as a mixture of them. The data are expressed in terms of three sets of variables:

- Documents: $d \in D = \{d_1, ..., d_N\}$, which are the observed variables. Let $N$ be the size of a given corpus.
- Words: $w \in W = \{w_1, ..., w_M\}$, which are the observed variables. Let $M$ be the number of distinct words in the corpus (i.e., the dictionary).
- Topics: $z \in Z = \{z_1, ..., z_K\}$, which are latent (or hidden) variables. The number $K$ has to be specified a-priori.

These variables are linked together in a graphical model (based on the aspect model) which associates the topics $z$ with the observed pairs $(d,w)$ as reported in figure 2.3. This also describes a generative process for the documents [4]:

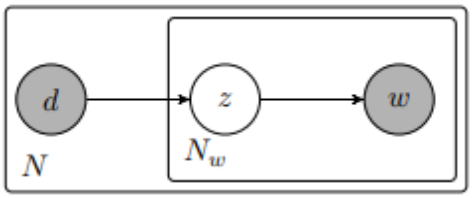



Fig. 2.3 The graphical model using plate representation for pLSA.

A generative process is described as follow:

- First, a document $d_n$ is random selected with probability $P(d)$.
- For each word $w_i i \in \{1,...,N_w\}$ in the document $d_n$:
  - Select random a topic $z_i$ from a multinomial conditioned on the given document with probability $P(z|dn)$.
  - Select random a word $w_i$ with probability $P(w|z_i)$ from a multinomial conditioned on the previous chosen topic.

Some important assumptions should be made for the described generative model:

- **Bag-of-words representation**. Each document is regarded as an unordered collection of words. This means that the joint variable *(d, w)* is independently sampled and, by this way, the joint distribution of the observed data will be factorised as a product $P(D,W) = \prod_{(d,w)} P(d,w)$.
- **Conditional independence**. Given a topic $z$, words and documents are conditionally independent: $P(w,d|z) = P(w|z)P(d|z)$ or $P(w|d,z) = P(w|z)$

The model can be completely defined by specifying the *joint distribution*. Using the product rules and the conditional independence assumption, we obtain

$$P(w|d) = \sum_{z \in Z} P(w|z)P(z|d)$$

$$P(w,d) = \sum_{z \in Z} P(z)P(d|z)P(w|z)$$

Thought the *likelihood maximisation*, the parameters can be estimated, by finding those values that maximise the predictive probability for the observed word occurrences. This is a non-convex optimisation problem and it can be solved by using *Expectation-Maximisation* (EM) algorithm for the log-likelihood [106].

### Latent Dirichlet Allocation

The *Latent Dirichlet Allocation* (LDA) is one the most famous and most used probabilistic topic modelling algorithm. It is a Bayesian method for topic extraction in a collection of documents. The intuition behind LDA is that documents are mixtures of multiple topics [29]. Topics are defined to be distributions over a fixed vocabulary. Documents, instead, are seen as a distribution over the set of different topics, thus showing multiple topics in different proportions. Thus, the LDA algorithm models the given textual dataset with a document-topics and a topic-terms probabilities distribution. LDA can be used to infer the topic hidden in a textual dataset and it estimates the parameters in the topic-terms and document-topics distributions using Markov chain Monte Carlo (MCMC) simulations [111]. As most of the topic modelling algorithms, LDA requires the number of topics to be previously known and defined. However, finding the optimal number of topic value that have to be discovered using the LDA is not trivial, and it is a open research issue in the scientific community [4].

The Latent Dirichlet Allocation (LDA) is a generative probabilistic model for collections of discrete data such as text corpora [42]. LDA is a Bayesian method for topic extraction in a collection of documents. The goal of topic modelling is to automatically discover the topic from a collection of textual data.

The graphical representation of the model is reported in Figure 2.4. Each node represents a random variable, in particular hidden nodes (e.g. topic proportions, assignments, and topics) are unshaded, while observed nodes (i.e., words of the documents) are shaded. Rectangles indicates replication, $N$ denotes words collected in the documents, while $D$ indicates the documents in the corpus.

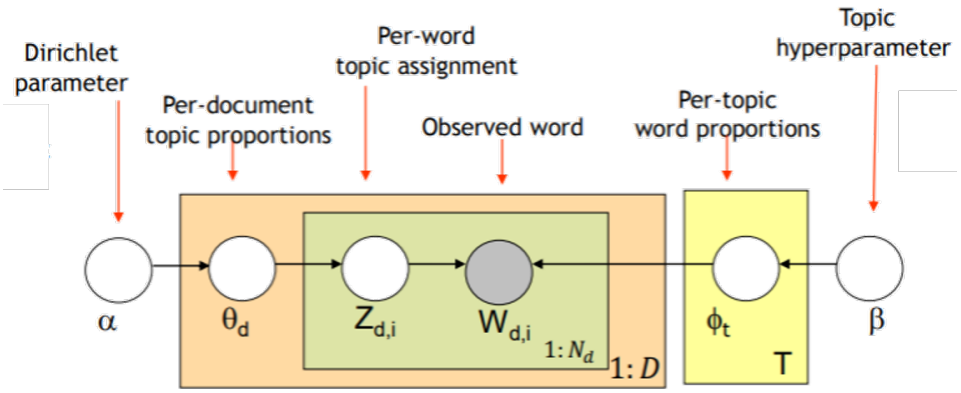



Fig. 2.4 The graphical model for latent Dirichlet allocation.

Using Bayesian inference (posterior inference) LDA infers the hidden structure to discover topics inside the collection under analysis. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterised by a distribution over words. LDA is a mathematical method for estimating both of these at the same time: finding the mixture of words that is associated with each topic, while also determining the mixture of topics that describes each document. It treats each document as a mixture of topics, and each topic as a mixture of words. A topic is formally defined as a distribution over a fixed dictionary. However, these topics are specified a-priori (technically, the LDA model assumes that the topics are generated first, before the documents). Then, for each document in the collection, words are generated through a two-stage process:

1. A distribution over topic is randomly chosen.
2. For each word in the document.
    a) A topic is randomly chosen from the distribution defined at the previous step (Step 1).
    b) A word is randomly chosen from the corresponding distribution over the dictionary.

This simple intuition highlights that documents present more topics. However, each document exhibits topics in different proportions (step 1); then each word in each document is drawn from one of the topics (step 2b), where the selected topic is chosen from the per-document distribution over topics (step 2a). Therefore, all the

documents in the corpus share the same set of topics, while the proportions of these topics in which each topic is exhibited are different.

The name latent Dirichlet allocation is due to the fact that the distribution that is used to randomly draw the per-document topic distributions in step 1 is called Dirichlet distribution. In the generative process for LDA, the result of the Dirichlet is used to allocate the words of the document to different topics. However, the documents themselves are observed, while the topic structure (i.e., the topics, document-topic distribution, and the topic-word distribution) is a hidden structure.

The LDA modelling describes topics and words as probabilistic distributions from which the document terms will be drawn. Documents are then seen as a distribution over a mixture of latent topics, since each term of a document is drawn from the vocabulary taking into account the terms' probabilities for each given topic of the document's mixture [29]. Specifically, to generate each document in the corpus, the steps performed are:

1. Choose random the number of terms from a Poisson distribution;
2. For each of the document's words:
   - Choose random a topic $z_n$ from Multinomial($\boldsymbol{\theta}$), where $\boldsymbol{\theta}$ is a Dirichlet($\boldsymbol{\alpha}$), representing the document-topics distribution;
   - Choose random a word $w_n$ from Multinomial($\boldsymbol{\phi}_{z_n}$), where $\boldsymbol{\phi}$ represents the topic-words distribution ($\boldsymbol{\phi} \sim$ Dirichlet($\boldsymbol{\beta}$)), conditioned on the topic $z_n$.

Hence, the joint multivariate distribution for the whole corpus of the document-topics distribution $\boldsymbol{\theta}$, the set of $K$ topics $\mathbf{z}$ and the set of $N$ terms $\mathbf{w}$ is defined as:

$$p(\mathscr{D}|\boldsymbol{\alpha},\boldsymbol{\beta}) = \prod_{d=1}^{K} \int p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \left( \prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_n|\boldsymbol{\theta}_d) p(w_{dn}|z_{dn},\boldsymbol{\beta}) \right) d\boldsymbol{\theta}_{\mathrm{d}},$$

where

- $\boldsymbol{\alpha}$ represents the concentration for the prior placed on documents' distributions over topics ($\theta$); This means that low $\alpha$ values will create documents that likely contain a mixture of only few topics, while high values will place more weight on having documents composed of many dominant topics.

- $\boldsymbol{\beta}$ describes the concentration for the prior placed on topics' distributions over terms. This means that low $\beta$ values will likely produce topics that are well described just by few words, while high values will create topics composed by a mixture of most of the words (and so not any word specifically).

With a corpus of documents $X$ in input, the generative LDA model can then be used to support inference of the posterior distribution of the latent variables for the given corpus. Generally, computing these distributions it is unfeasible, and so it is impossible to exactly solve this posterior Bayesian inferential problem. To overcome this problem, several approximate inference algorithms have been proposed in literature. For further details see Chapter 3.

**State-of-the-art strategies to automatically configure the joint-approach model parameters.**

The weighted matrix $X$, which represents the documents in the corpus, is reduced in size using LSA (Latent Semantic Analysis). For a given reduction factor, LSA allows to reduce the dimensionality of the matrix without losing significant information. However, choosing the most proper reduction factor is not a trivial task. To reduce the dimensionality of the matrix and uncover the hidden concepts in the dataset, LSA applies the SVD transformation (Singular Value Decomposition) to $X$, in order to decompose the matrix into the product of its components.

First, a reduction parameter for the algebraic model should be set. The main challenge to LSA has been the alleged difficulty in determining the optimal number of dimensions to use for performing the SVD, which is a crucial aspect of many text mining solutions. The issue with using SVD is to determine how many dimensions, and so, how many concepts, should be set when the matrix is approximated. Too few dimensions and important reasons are left out, too many and the noise caused by random choices of words will creep up again [5].

The actual number of dimensions that can be used is limited by the number of documents in the collection. Research has demonstrated that around 300 dimensions will usually provide the best results with moderate-sized document collections (hundreds of thousands of documents) and perhaps 400 dimensions for larger document collections (millions of documents) [112]. However, recent studies indicate that 50-1000 dimensions are suitable depending on the size and nature of the document collection [41].

In [5, 113], the authors plot the singular value in a scree plot. While the author in [113] analyse the amount of variance in the data after computing the SVD to select the optimal number of dimensions to retain, in [5] the authors analyse the variation between each singular value and the following one.

For the PCA several strategies have been proposed. As described before, the variance contained in the data could be used to select a threshold for the number of components (i.e., the scree plot analysis). The analyst should select the dimensionality associated with the knee of the curve as the cut-off point for the number of dimensions to retain. Others argue that some quantity of the variance must be retained, and the amount of variance in the data should dictate the proper dimensionality to retain. Seventy percent is often mentioned as the amount of variance in the data that should be used to select the optimal dimensionality for recomputing the PCA [114, 115].

However, the proposed methodologies might lead to an incorrect choice because a local minimum can be met. To overcome this issue, we propose a new algorithm whose description is reported in Chapter 3.

After the reduction phase that improve the complexity of the text analysis, a partitional clustering algorithms is applied to discover homogeneous groups of documents with a similar topic. In literature, one of the most popular clustering algorithms is the K-Means [116] algorithm capable to identify the cluster set in a limited computational time by producing quite good results in many application domains. Each group is represented by its centroid computed as the average of all the objects in the cluster. One of the biggest drawbacks of K-Means is that it requires the number of clusters to be a-priori specified. To address this issue, in literature the trend of the SSE quality index is analysed [105]. The optimal value of $K$ must be selected at the coordinates where the marginal decrease in the SSE curve is maximised. The SSE index [105] measures the cluster quality in terms of cluster cohesion. It is computed as the total sum of squared errors for all objects in the collection, where for each object the error is computed as the squared distance from the closest centroid. However, the SSE method usually sets a lower value for the desired number of clusters, as a local minimum can be met.

**State-of-the-art strategies to automatically configure the probabilistic model parameters**

LDA can be used to infer the topic hidden in a textual dataset. However, as most of the topic modelling algorithms, LDA requires the number to topics to be previously

known and fixed. However, finding the optimal values for the number of topics that have to be discovered by the LDA is not trivial, and it is instead an open issue in the scientific community.

Two of the most well-known state-of-the-art algorithms to automatically determine the optimal number of topics have been discussed: (i) the **Rate of Perplexity Change** and (ii) the **Entropy optimised Latent Dirichlet Allocation**. These methods are based on different approaches: the first one is based on the variation of the average perplexity; while the second one is based on the entropy contained in the LDA model.

In many clustering algorithms finding the optimal value for the number of topics $K$ is not trivial, as described during the joint-approach. Thus, the number of topics has great influence on the results of the clustering process, but often the evaluation of the results is subjective, difficult to be interpreted and time-consuming.

This model parameter has to be set carefully, since based on it the number of clusters, and so the final clustering result, will drastically change. Indeed, too low $K$ values would lead LDA to be too coarse to be able to identify proper clusters, while $K$ values that are too big would lead to a very complex model, difficult to be interpreted and difficult to be validated.

The *Rate of Perplexity Change* (*RPC*), proposed by Zhao et al. [117], is a heuristic approach aiming to estimate the best value for the number of topics $K$ for the LDA model. Proposing this strategy, the authors wanted to overcome the problems of perplexity, the methodology originally proposed to evaluate the LDA models and determine which clustering statistically better describes a given dataset. According to [29], the lower the perplexity of a model, the better it performs describing the data collection (a detailed description of perplexity as evaluation index is given in Chapter 3). Finding this approach not stable and too varying even for the same dataset, the RPC strategy aims to outperform the perplexity approach.
This method, claimed to be stable and effective, considers how the variation of the average perplexities $P_i$ for $K$ candidate number of topics ($P_1$, ..., $P_i$, ... $P_K$) changes with respect to the topic's numbers $K_i$ ($1 < i \leq K$) [117]. The equation of the RPC index is:

$$RPC(i) = \left| \frac{P_i - P_{i\text{-}1}}{K_i - K_{i\text{-}1}} \right| \tag{2.2}$$

Given the definition of the RPC function, the first change point of the RPC curve, i.e. the first $i$ that satisfies the equation $RPC(i) < RPC(i+1)$, is chosen to be the most suitable value for the number of topics $K$.

The *Entropy optimised Latent Dirichlet Allocation* (*En-LDA*) [118], is an entropy-based measure to optimally select the number of topics needed to properly describe a corpus using the LDA topic modelling.
Entropy measures the average information contained in an event. Generally, information can also be seen as the uncertainty characterising the probabilistic event itself: in our case, considering the probabilistic model generated by the LDA, entropy represents the uncertainty of the model when describing the dataset under analysis. The idea of the En-LDA authors is to measure the entropy of different LDA models obtained with different configurations, to assess which of them is likely to be the better one. To do that, they measured the entropy of each term given a document $d_m$ using topics as the probabilistic labels of the word. Considering all the documents in a corpus, the entropy of all the words are then aggregated to estimate the overall entropy of the terms given the distribution of words with respect to the topics and the distribution of the topics with respect to the documents.
The overall entropy of the clustering is then measured as:

$$Entropy(K) = \sum_{m=1}^{M} \sum_{k=1}^{K} p(z_m = k|d = d_m) \left( \sum_{n=1}^{N_m} -p_{t,k,m}\, ln(p_{t,k,m}) \right) \tag{2.3}$$

where $M$ is the number of documents in the corpus, $K$ is the number of topics in the clustering, $N_m$ is the number of terms in document $d_m$, $z_m$ the document's topic and $z_n$ the topic of the word $w_n$. $p_{t,k,m}$ is instead the normalised probability of the word $w_n = t$ with respect to the topic $z_n = k$. In other words, $p_{t,k,m}$ represents the probability that the term $w_n$ is $t$ under the $k$th topic. In formula, this is expressed by: $p_{t,k,m} = \frac{p(w_n = t|z_n = k)}{\sum_{n=1}^{V_m} p(w_n = t|z_n = k)}$, having $V_m$ equal to the size of the vocabulary for the document $d_m$.
To choose the most proper $K$ number of topics, several LDA models have to be computed for different values of $K$, and then the one with the minimum value of entropy is selected to be the optimal number of partitions to describe the data collection.
The approach is considered to be stable and able to handle very low and very high number of clusters. Indeed, very low $K$ values lead to very large entropy values, as

well as very large number of clusters.

## 2.4 Visualisation

Text Mining has become quite mainstream nowadays as the tools to make a reasonable text analysis are ready to be exploited and give nice and reasonable results. The combination of text mining and visualisation tools can make this process even more effective. Text visualisation has become a growing and increasingly important sub-field of information visualisation [119].

In the literature, different kinds of visualisation to easily navigate data and knowledge have been proposed. The methods can be classified into the following categories [120]:

- **Trend chart.** Trend charts are also known as run charts, and are used to show trends in data over time. They include bar charts or pie charts able to describe the overall content of data. Specifically, for the text mining domain, different visualisations are exploited to characterise the overall distribution of documents in each corpus. Unlike plain text, data visualisation takes complex information and boils it down to a simple representation. Simple statistics can be plotted to analyse the frequency distribution of terms or the document length for each corpus. Moreover, these charts include examples for counting total and unique frequencies of words within a dataset.

- **Word list and word cloud.** The simplest and most common form of text visualisation is a tag (or word) cloud. Tags are arranged in space varied in size, color, and position based on tag frequency, categorisation, or significance. In [121], the author have created a series of visualisations highlighting the words being used in the speeches of both gatherings. These word-cloud-like word bubble clouds serve as a great interface for looking at the differences in the two conventions and for browsing through quotes from the talks.

- **Networks, relations and connections.** The visual graph explorer for graph visualisation, discovery and exploration of connections generates data visualisations of networks, connections and relations between entities. Different

visualisation techniques have been proposed in the state-of-the-art, including tags visualisation from the content of the documents [122] and graphs visualisation for the term-distribution of topics. In [123], the authors explore the knowledge graphs to represent texts in order to gain a better understanding of textual data and to tackle the dynamic nature of knowledge. The use of graphs to represent text and compute analysis is a well-known task; many of the approaches presented above focused on semantic relations between the words when representing texts as networks [124].

All of these approaches are helpful in gaining a better understanding of text. Another interesting extension is the inclusion of an extra layer of ontologies on top of the textual data. The decisions regarding which concepts are related together are based on their effective affinity, their causal relations, and semantic analysis.

## 2.5 Final consideration

Since both the joint and the probabilistic approach do not provide a stable and a resilient method to find a proper number of topics for the dataset under analysis, the main goal of this dissertation is to propose novel methods to determine an optimal number of topics (clusters) for both the topic detection and the clustering analysis. The entire text mining process has been integrated in a novel engine, named ESCAPE (**E**nhanced **S**elf-tuning **C**haracterisation of document collections **A**fter **P**arameter **E**valuation) which includes auto-selection strategies to off-load the end-user from the burden of the parameter tuning, achieving good qualitative results. ESCAPE is able to automatically analyse different types of textual collections, including also several innovative strategies to graphically show the discovered knowledge.

# Chapter 3

# Topic Modelling and document clustering

In this Chapter, the proposed and developed engine to automatically analyse collection of textual data is presented. The new engine, named ESCAPE (**E**nhanced **S**elf-tuning **C**haracterisation of document collections **A**fter **P**arameter **E**valuation), includes different algorithms to perform document clustering and topic modelling. Ad-hoc self-tuning strategies have been integrated in ESCAPE to automatically configure specific algorithm parameters, as well as it includes novel visualisation techniques and evaluation quality measures to analyse the performances of both methodologies.

The chapter is organised as follows. Section 3.1 introduces ESCAPE, a new engine able to suggest to the analyst possible good configurations for the complete data analytics pipeline to perform both cluster analysis and topic modelling of a collection of textual documents. In detail, ESCAPE includes three main building blocks which are (i) *Data processing and characterisation* (see Section 3.2), (ii) *Self-Tuning Exploratory Data Analytics* (see Section 3.3), and (iii) *Knowledge validation and visualisation* (see Section 3.4).

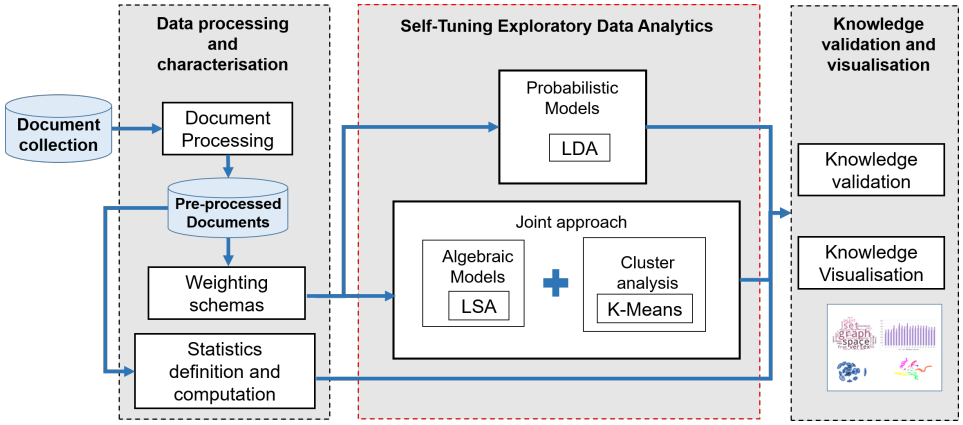



Fig. 3.1 The ESCAPE System Architecture.

## 3.1 ESCAPE

In this Section we describe our proposed engine, named ESCAPE (**E**nhanced **S**elf-tuning **C**haracterisation of document collections **A**fter **P**arameter **E**valuation), which includes automatic strategies to relieve the end-user of the burden of selecting proper values for the overall process of cluster collections of textual data.

The ESCAPE architecture, reported in Figure 3.1, includes three main components: (i) *Data processing and characterisation*, *Self-Tuning Exploratory Data Analytics*, and (iii) *Knowledge validation and visualisation*. It is also important to note, however, that the proposed engine ESCAPE is not necessarily tailored to text mining and has applications to other interesting problems involving collections of data, including data from several IoT domains and so on [8, 12, 13, 9]. Specifically, in [12] a data mining engine including both exploratory and unsupervised data analytics algorithms, devised to build transparent models correlating weather conditions and energy consumption in buildings. In [8] an extension of the engine is presented. First, a partitional clustering algorithm is applied to weather conditions. Then, resulting clusters are characterised by means of generalised association rules, which provide a self-learning explainable model of the most interesting correlations between energy consumption and weather conditions at different granularity levels. In [13], a multi-tiered data mining engine to discover interesting knowledge items from real pollutants measurements collected in a major Italian city is developed, while in [9] a framework tailored for the analysis of the Energy Performance Certificates collected in a major Italian region is presented. All these works include self-tuning strategies

to automatically configure the exploratory phase, reducing the end-user intervention in the parameter tuning.

Clearly, the first step in text mining is to collect textual data (i.e., a set of documents of interest). In many text extraction scenarios, documents may already be provided or may be part of the problem description. For example, a Web page retrieval application specifies relevant documents such as a set of Web pages on the Intranet. Sometimes, documents can be obtained from databases or data warehouses. In some applications, it may be necessary to have a data collection process. For example, for a Web application that includes a number of stand-alone Web sites, a Web crawler [125] can be exploited to collect documents. Sometimes the document collection could be extremely large and data sampling techniques can be used to select a manageable set of relevant documents. These collections are usually called the document corpora.

### 3.1.1 Notation and terminology

In this subsection, the main notation used throughout the dissertation are reported. This is useful in that it helps to guide intuition, particularly when the self-tuning methodologies will be introduced. Formally, the following terms have been defined:

- A *word* is the basic unit of discrete textual data which is defined as an item from a vocabulary indexed by 1,...,V.
- A *document* is a sequence of $N$ words denoted by $d = (w_1, w_2, ..., w_N)$, where $w_i$ is the $i^{th}$ word in the sequence.
- A *corpus* is a collection of $D$ documents denoted by $D = \{d_1, d_2, ..., d_D\}$.

## 3.2 Data processing and characterisation

Once the documents are collected, they have to be properly pre-elaborated. Pre-processing is an important and critical task that affects the quality of the text mining results. It includes many steps described below.

### 3.2.1 Document processing

Text pre-processing is an essential part of any Natural Language Process (NLP) system, since the characters, words, and sentences identified at this stage are the fundamental units passed to all further processing stages, from analysis and tagging components, such as morphological analysers and part-of-speech taggers, through applications, such as information retrieval and machine translation systems. It is a collection of activities/steps in which text documents are pre-processed. Because the text data often contains some special formats like number formats, date formats and the most common words that unlikely to help text mining such as prepositions, articles, and pro-nouns can be eliminated, since these words are not informative for any texts.

In the *Textual data processing components*, five steps are performed sequentially as interrelated tasks:

1. *document splitting*: documents can be split into sentences, paragraphs, or analysed in their entire content, according to the next analytics task. Short documents like emails or social posts (e.g. tweets) are naturally translated into a single vector for each message. Longer documents instead can be analysed as the entire document or can be broken up into sections or paragraphs. Choosing the proper scope depends on the goals of the text mining task: for clustering task (as the scope of this paper) the entire document is analysed in its entire content; for sentimental analysis, document summarisation, or information retrieval, smaller units of text such as paragraphs or sections might be more appropriate;

2. *tokenisation*: is the process of segmenting a text or texts into tokens (i.e., words) by the white space or punctuation marks within the same split;

3. *case normalisation*: Capitalisation helps human readers differentiate, for example, between nouns and proper nouns and can be useful for automated algorithms as well. In many analytics tasks, however, an upper-case word at the beginning of the sentence should be treated no differently than the same word in lower case appearing elsewhere in a document. This step converts each token to completely upper-case or lower-case characters;

4. *stemming*: each token is mapped into its own root form. It includes the identification and removal of prefixes, suffixes, and pluralisation;

5. *stopwords removal*: stopwords are the grammatical words which are irrelevant to text contents (e.g. articles, pronouns, prepositions), so they need to be removed for more efficiency. These common words can be discarded before the feature generation process.

These activities represent each split via the Bag-Of-Word (BOW) representation, a set of terms generated by disregarding grammar and even word order but representing the document's main themes. Frequency information on the word counts can be quite useful in reducing dictionary size and can sometimes improve predictive performance for some methods. The most frequent words are often stopwords and can be deleted. The remaining most frequently used words are often the important words that should remain in a local dictionary. The very rare words are often typos and can also be dismissed. In general, the smaller the dictionary, the greater the intelligence to capture the most and the best words [126]. The use of tokens and stemming are examples of useful procedures in the composition of smaller dictionaries. Once the set of words has been determined, the document collection can be converted to matrix structure format.

### 3.2.2 Statistics definition and computation

ESCAPE includes the computation of several statistical indices [3, 5, 127] to characterise the document collection data distribution:

- *# categories*: the number of topics/clusters in the textual collection under analysis (if known a-priori);
- *Avg frequency terms*: the average frequency of token occurrence in the corpus;
- *Max frequency terms*: the maximum frequency of token occurrence in the corpus;
- *Min frequency terms*: the minimum frequency of token occurrence in the corpus;

- *# documents*: the number of textual documents in the corpus (i.e., total number of splits defined by the analyst);
- *# terms*: number of terms in the corpus, with repetitions (i.e., all words of a textual collection);
- *Avg document length*: the average length of documents in the corpora;
- *Dictionary*: the number of different terms in the corpus, without repetition (i.e., all words that are different from each other in a textual collection);
- *TTR*: the ratio between the dictionary variety (*Dictionary*) and the total number of tokens in a textual collection (*# terms*);
- *Hapax %*: the percentage of Hapax, which is computed as the ratio between the number terms with one occurrence in the whole corpus (Hapax) and the cardinality of the Dictionary;
- *Guiraud Index*: the ratio between the cardinality of the Dictionary and the square root of the number of tokens (*# terms*). It highlights the *lexical richness* of a textual collection.

The jointly analysis of these statistical features is able to describe the lexical richness and characterise the data distribution of each collection under analysis. ESCAPE includes also a Boolean feature, named *remove-Hapax*, which if it is set to *True*, ESCAPE removes the Hapax words for subsequent analyses, otherwise these words are included in the analysis. This step could lead different results for the different strategies included in ESCAPE. Indeed, algebraic models are less influenced by the presence of Hapax, as in the decomposition their affection is overridden by the most frequent terms. Probabilistic models, on the other hand, are influenced in a more negative way, as they introduce noise within the creation of the model.

### 3.2.3 Term relevance

This ESCAPE's step entails representing a corpus through several weighting functions to highlight the relevance of specific words in the document collection. The weight of each word represents its importance degree. Different weighting schemas are expected to lead to different results. Based on the document statistical features

and the desired granularity of the outcomes, one of the weighting schema is expected to outperform with respect to the others.

Let $D = \{d_1, d_1, \ldots, d_{|D|}\}$ be a corpus of documents, named corpus, and $V = \{t_1, t_2, \ldots, t_{|V|}\}$ the set of distinct terms in the textual collection, i.e. the set of all tokens used at least once in a document. $D$ is represented as a matrix $X$, named *document-term* matrix, in which each row corresponds to a document in the collection and each column, one for each $t_j \in V$, corresponds to a term in the vocabulary.

These cells represent the presence of the dictionary's words in a document collection. To measure the relevance of terms appearing in the document, each cell in the matrix $D$ is associated with a *weight*. A weight $x_{ij}$ is a positive real number associated with each term $t_j$ of $d_i$, and quantifies its level of importance. Various weighting functions, combining a local term weight with a global term weight, have been proposed in [70]. A weighting function applied on a collection $D$ generates its weighted matrix $X$. Specifically, for each term $t_j$ of a document $d_i$ the corresponding weight $x_{ij}$ in $X$ is computed as the product of a local term weight ($l_{ij}$) and a global term weight ($g_j$) ($x_{ij}$=$l_{ij} * g_j$). A local weight $l_{ij}$ measures the relative frequency of a specific term $j$ in a particular document $i$, while the global weight $g_j$ describes the relative frequency of the specific term $t_j$ within the whole corpus $D$.

ESCAPE includes three local term weights: *Term-Frequency* (TF) [68], *Logarithmic term frequency* (Log) [69] and *Binary* (Boolean) [70] and three global term weights: *Inverse Document Frequency* (IDF) [68], *Entropy* (Entropy) [69] and *Term-Frequency* ($\text{TF}_{glob}$) [70]. They are defined in Table 3.1.

The TF weight, defined as $tf_{ij}$, represents the frequency of term $j$ in document $i$, while the Log weight evaluates the term frequency in base-2 logarithmic scale, which is used to diminish the large number frequencies. While the binary weight function is equal to 1 if the frequency was non-zero and 0, otherwise. Intuitively, the first two local weights give increasing importance to more frequent words, but the logarithmic gives progressively smaller additional emphasis to larger frequencies, while the third measure is sensitive only to whether the word is in the document. The next step beyond counting the frequency of a word in a document is to modify the count by the perceived importance of that word.

All of the global weighting schemas basically give less weight to terms that occur frequently or in many documents. The ways in which this is done involve interesting variations in the relative importance of local frequency, global frequency, and

| Weight | Definition |
|---|---|
| Local | TF = $\text{tf}_{ij}$ |
| | LogTF = $\log_2(\text{tf}_{ij}+1)$ |
| | Boolean = $\{0, 1\}$ |
| Global | IDF = $\log \frac{\lvert D\rvert}{\text{df}_j}$ |
| | Entropy = $1+\sum_i \frac{p_{ij}\log p_{ij}}{\log n}$ |
| | $\text{TF}_{glob}$ = $\text{tf}_j$ |

Table 3.1 Local and Global weight functions exploited in ESCAPE.

document frequency. In particular, the global weight IDF measures the rareness of a term and it is defined as the logarithm of the ratio between the number of documents in the corpus ($|D|$) and the number of documents in which term $j$ appears ($df_j$). The IDF of a rare term is high, whereas the IDF of a frequent term is likely to be low. The global weight function referred to as Entropy represents the real entropy of the conditional distribution given that the term $i$ appeared. In documents, high normalised entropy is considered good and low normalised entropy is considered bad. Entropy is based on information theoretic ideas and is the most sophisticated weighting schema. The assigns minimum weight to terms that are equally distributed over documents (i.e. where $p_{ij} = 1/ndocs$), and maximum weight to terms which are concentrated in a few documents. Entropy takes into account the distribution of terms over documents.

We explored in ESCAPE the effects of seven different term weighting schemas in each of the test collections. We performed analyses using: combination of three local weights discusses above (TF, LogTF and Binary) and each of the two global weights (Idf, Entropy), and one combination of a local Binary weight and a global $\text{TF}_{glob}$ weight. For example, combining the local weight TF with the global weight IDF tends to filter out common terms. More specifically, the TF-IDF weight $x_{ij}$ for the pair $(d_i, t_j)$ is high when term $t_j$ appears with high frequency in $d_i$ and low frequency in the collection $D$. When term $t_j$ appears in more documents, the ratio inside the IDF's log function approaches 1, and the IDF value of $t_j$ and TF-IDF weight ($x_{ij}$) become close to 0. Instead LogTF-IDF penalises frequent words more than TF-IDF. All these combinations are analysed to show how the different schemas are able to characterise the same dataset at different granularity level.

## 3.3 Self-Tuning Exploratory Data Analytics

Document clustering and topic modelling are two closely related tasks which can mutually benefit each other [128]. Topic modelling can project documents into a topic space which facilitates effective document clustering. Cluster labels discovered by document clustering can be incorporated into topic models to extract local topics specific to each cluster and global topics shared by all clusters. In this Section, two well-known approaches for document clustering and topic modelling have been integrated. For each strategy, a brief description is reported, together with its main drawbacks and our proposed methodology to automatically select few good values for the entire analysis. Specifically, in Subsection 3.3.1, we reported the Joint-Approach, including our proposed algorithms to automatically suggest suitable values for the data reduction phase together with proper values for the clustering phase. While, in subsection 3.3.2, the probabilistic model LDA will be presented, including our new approach for discovering good partitions of a document collection.

### 3.3.1 Joint-approach

The joint-approach includes a data reduction phase computed through the Latent Semantic Analysis based on the Singular Value Decomposition, before the exploitation of the partitional K-Means algorithm. Below, a brief description of the two algorithms is reported, including the main drawbacks. Lastly, the Subsection ends with the two proposed self-tuning algorithms to choose suitable values for the reduction phase and the clustering analysis, respectively.

#### Latent Semantic analysis

To make the cluster analysis problem more effectively tractable, ESCAPE includes natural language process named LSA (Latent Semantic analysis) [41] [73]. LSA allows reducing the dimensionality of matrix $X$ while disregarding some irrelevant dimensions [75]. The choice of the correct dimensionality reduction, without losing significant information, is an open research issue and a very complex task. LSA is able to analyse relationships between groups of documents and terms generating sets of concepts in the corpus under analysis. Through the application of the Singular Value Decomposition (SVD), ESCAPE finds the hidden concepts. Too few dimen-

sions after the LSA process will lead to poor data representation, whereas too many dimensions will result in more noisy data. LSA arose from the problem of how to find relevant documents from search words. The fundamental difficulty arises when words are compared to find relevant documents, because what should be compared is the meanings or concepts behind the words. LSA attempts to solve this problem by mapping both words and documents into a concept-space and the comparison is done in this new space. In order to make this problem more effective tractable, some simplifications are introduced:

- *Documents* are represented as *bags of words*, where the order in which the words appear in the documents is not important. Only the frequency is relevant to measure the weight of terms in the corpus.
- *Concepts* are represented as patterns of words that appear together in the collection.

SVD is a matrix factorisation method that decomposes the original matrix (document-term matrix) $X$ into three matrices ($U$; $S$; $V^T$). $U$ is a $d \times r$ column-orthonormal matrix (i.e., $U^T U = I$), $S$ is a d×d diagonal matrix and $V$ is a $r \times t$ column-orthonormal matrix (i.e., $V^T V = I$). S is also called the concept-matrix, while U and V are called document-concept similarity matrix and term-concept similarity matrix, respectively. Each cell of the weighted matrix X is represented as $x_{ij} = \sum_{c=1}^{r} d_{i,c} \lambda_c t_{c,j}$, where each weighted term $t_i$ in document $d_j$ is expressed as a linear combination of term-concept and document-concept weights. We obtain the exact decomposition (lossless representation) of the original matrix in Equation 3.1.

$$X = USV^T \tag{3.1}$$

The matrix $S$ includes a singular value for each dimension (term) in the document collection under analysis. The significance of each dimension is represented by the magnitude of the corresponding singular value in $S$. Through the SVD decomposition some insignificant dimensions in the transformed space can be easily identified to approximate (in the least square sense) matrix $X$. Insignificant dimensions in $S$ represented by a low magnitude of singular values may represent noise in the data and should be disregarded in the subsequent analysis steps. The singular values model the relative importance of the dimensions. $r$ is the rank of the original matrix

X and the singular values determine the relative importance of the dimensions. As the singular values decrease, so does the effect of the dimension. The $k$ selected singular values correspond to the hidden concepts.

Since both U and V are orthonormal, we can multiply both the side of equation 3.1 by V, we obtain $XV = US$. It can be seen as a projection of documents in the r-dimensional concept space. In this new space, documents are represented by the row of $US$. Given a target dimensionality $k$ ($k << r$), it is possible to obtain an optimal approximation of the original matrix by retaining only the k largest singular values of the matrix S. Among all the rank-k approximation, we analyse the one that minimises the Frobenius norm. However, LSA has no theoretical optimal reduced dimension, and its computational estimation is difficult without the potentially expensive process of trying many test cases.

To identify the main relevant dimensions ($K_{LSA}$) in $X$, ESCAPE includes an innovative algorithm (named ST-DARE, see subsection 3.3.1 for more details). Given $K_{LSA}$, ESCAPE uses only the largest singular $K_{LSA}$ values in $S$ and sets the remaining ones to zero. The approximated matrix of $X$, denoted $X_{K_{LSA}} = U_{K_{LSA}} S_{K_{LSA}} V_{K_{LSA}}^T$ is obtained by reducing all three decomposed matrices $(U, S, V^T)$ to rank $K_{LSA}$. In general, the low-rank approximation of X by $X_{K_{LSA}}$ can be viewed as a constrained optimisation problem with respect to the constraint that $X_{K_{LSA}}$ have rank at most $K_{LSA}$. When forced to squeeze the terms-documents down to a k-dimensional space, the SVD should bring together terms with similar co-occurrences. This intuition suggests that the dimensionality reduction could improve the results [129].

### K-Means Algorithm

In the joint-approach, the singular value decomposition is applied to data to reduce the dimension of the data prior to the learning process using the K-Means Clustering. The different document-concept vectors could also be clustered using a clustering algorithm such as K-Means. The difference between clustering and LSA is that clustering algorithms assign each document to a specific cluster, while LSA assigns a set of topic loadings to each document. However, a K-Means algorithm applied after the singular value decomposition improve the results, as shown in [5, 6, 3, 127]. As a matter of fact, the large dimensions of data become obstacles.

K-Means is one of the simplest unsupervised learning algorithms that solve the wellknown clustering problem [104]. It is a simple and partitional strategy that attempts to find $K$ clusters, represented by their centroids, given by the mean value of the objects (i.e., textual documents in this thesis) in each cluster. Initially, the partitional algorithm randomly chooses $K$ documents of the collection as centroids. Then each document is assigned to the cluster whose centroid is the nearest to that document. Finally, the mean of all the documents in each cluster is computed to recalculate the new centroids. The process iterates until the centroids do not change. Unlike other algorithms (e.g. hierarchical clustering), K-Means is computationally faster and produces tighter clusters, especially if clusters are globular. However, K-Means requires the a-priori knowledge of the number of clusters, which is usually hard to define [105]. The similarity between two documents is usually measured according to a notion of similarity/distance in the space describing the document terms. Although the cosine similarity is the most common similarity measure exploited, as discussed in [15], the Euclidean distance can also be used after normalising the document vectors with respect to the Euclidean norm. Thus, the Euclidean distance is usually exploited to measure the distance among documents. As a matter of fact, for normalised vectors cosine similarity and Euclidean similarity are connected linearly.

Cosine distance is actually cosine similarity [130, 131] and it is computed as $\cos(\mathbf{x},\mathbf{y}) = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2 \sum y_i^2}}$, where $x$ and $y$ are two vectors. With Euclidean distance for normalised vectors we obtained (i.e., $\sum x_i^2 = \sum y_i^2 = 1$):

$$||\mathbf{x}-\mathbf{y}||^2 = \sum (x_i - y_i)^2 = \sum (x_i^2 + y_i^2 - 2x_i y_i) =$$

$$\sum x_i^2 + \sum y_i^2 - 2\sum x_i y_i = 1 + 1 - 2cos(\mathbf{x},\mathbf{y}) = 2(1 - \cos(\mathbf{x},\mathbf{y})).$$

Note that for normalised vectors $\cos(\mathbf{x},\mathbf{y}) = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2 \sum y_i^2}} = \sum x_i y_i$. This demonstrates that there is a direct connection between these two distances for normalised vectors. After turning each vector into a unit vector, the partitional algorithm is applied.

ESCAPE entails the discovering of groups of documents with a similar topic through the self-assessment of the quality of the discovered clusters. It includes an algorithm to automatically configure the cluster analysis activity through the analysis of different quality metrics to evaluate the obtained partitions. To this aim, several configurations have been tested by ESCAPE, varying the specific-algorithm parameter (i.e., number of desired clusters).

### Self-Tuning Data Reduction algorithm

The ST-DaRe (Self-Tuning Data Reduction) algorithm in ESCAPE automatically selects three good values for the LSA parameter to identify a good number of dimensions to consider in the subsequent analytics steps without losing significant information. The correct choice of the number of dimensions to be considered is an open research issue [70]. A simple approach is to identify the maximum decrease point in the singular value curve. However, it might lead to an incorrect choice because a local minimum can be met.

The original ST-DaRe algorithm [5] uses three parameters experimentally set (i.e., two thresholds and the singular value step) to analyse the variability of the singular value curve, i.e., plot of the singular values (in descending order) obtained through the SVD decomposition. The singular values are analysed in pairs using the predetermined singular value step defined by the end-user. Then, the marginal decrease in the curve is computed for each couple of singular values. If this decrease is comparable with either one of the two thresholds, or their average, the smallest singular value of the pair is selected as one of the three values. The pseudo-code is reported in Algorithm 1. This first version requires quite a number of parameters that are fixed to values achieving good results. To improve the proposed methodology, a self-tuning enhanced version has been proposed. In this way, only one input parameter is required, which analyses the trend of singular values in terms of their significance.

In ESCAPE, we include an enhanced version of ST-DaRe with only one input parameter to analyse the trend of singular values in terms of significance [6]. The significance of each dimension is represented by the magnitude of the corresponding singular value. Insignificant dimensions represented by a low magnitude of singular values may represent noise in the data and should be disregarded in the subsequent analytics steps. Thus, we only consider the first $T$ singular values for the analysis. Specifically, the mean and the standard deviation values of the magnitude of the first $T$ singular values are computed and then a confidence interval is defined. The selected three-good values of the number of dimensions to consider for the next analytics steps are distributed along the curve: (i) the first is the singular value in correspondence of the mean position, (ii) the second is the singular value in correspondence of the mean plus the standard deviation position, and (iii) the last one is the singular value in correspondence of the mean position of the previous ones. Through this method the problem of the local optimality choice is overcome.

**Algorithm 1:** The ST-DaRe pseudo-code

**Input :** $X, th_1, th_2, step$
**Output :** $K_{LSA}[3]$

$N = 0$;
// compute the SVD decomposition of matrix X;
$[U,S,V] \leftarrow X.computeSvd(X.numCols)$;
$s \leftarrow normSingularValues(S)$;
**if** $th_1 < th_2$ **then**
    $swap(th_1, th_2)$;
**end**
$f_1 \leftarrow false$; $f_2 \leftarrow false$; $f_3 \leftarrow false$;
**for** $i \leftarrow 0$ **to** $s.numCols - step$ *OR* $N = 3$ **do**
    // compute the marginal decrease;
    $\Delta \leftarrow s(i) - s(i+step)$;
    **if** $!(f_3)$ *AND* $\Delta < th_2$ **then**
        $K_{LSA}.push(i+step)$;
        $f_3 \leftarrow true$;  N++;
    **else if** $!(f_2)$ *AND* $\Delta < (th_1 + th_2)/2$ **then**
        $K_{LSA}.push(i+step)$;
        $f_2 \leftarrow true$;  N++;
    **else if** $!(f_1)$ *AND* $\Delta < th_1$ **then**
        $K_{LSA}.push(i+step)$;
        $f_1 \leftarrow true$;  N++;
    **end**
    $i \leftarrow i+1$;
**end**
**if** $K_{LSA}.length < 1$ **then**
    // take only values greater than 1, when there is not a clear bend;
    $K_{LSA}.push(values\_Greater\_Than\_One(S).length)$
**end**

$T$ at most will be equal to the rank of the document-term matrix. However, in our framework we have set this value equal to 20% of the rank. Since the number of documents for all the textual corpora analysed is much smaller than the vocabulary used in each collection, the value $T$ is set by ESCAPE to the 20% of the number of documents. The proposed enhanced version is reported in Algorithm 2.

**Algorithm 2:** The Enhanced ST-DaRe pseudo-code

**Input :** $X$, $T$
**Output :** $K_{LSA}[3]$

$N = 0$;
// compute the SVD decomposition of the truncated matrix X;
$[U,S,V] \leftarrow X.computeSvd(T)$;
$s \leftarrow normSingularValues(S)$;
// compute the mean of singular values;
$mean = s.mean()$;
// compute the standard deviation of singular values;
$stand\_deviation = s.std()$;
// compute the three values;
$val1 = s[mean]$;
$val2 = s[mean + stand\_deviation]$;
$val3 = s[(val1 + val2)/2]$;
$K_{LSA}.push(val1, val2, val3])$

### Self-Tuning Clustering

In ESCAPE, the cluster analysis is addressed via the K-Means algorithm, which is a simple and partitional strategy that attempts to find $K$ clusters. However, it requires apriori knowledge of the number of clusters, which is usually complex to set. To address this issue, ESCAPE includes a Self-Tuning Clustering algorithm to automatically find a good value for the number of hidden topics in the textual corpus under analysis. To automatically compare and rank different document partitions obtained with different K-Means configurations, ESCAPE entails the discovering of groups of documents with a similar topic through the self-assessment of the quality of the discovered clusters. The proposed algorithm, automatically configure the cluster analysis activity through the analysis of different quality metrics to evaluate the obtained partitions. To this aim, several configurations have been tested by ESCAPE, varying the specific-algorithm parameter (i.e., number of desired

clusters). Once the $K$ clusters have been formed starting from the collection of textual documents, the clustering outcomes are subject to clustering validity assessment, by using three indicators based on the calculation of the widely used silhouette [132]. The *silhouette index* is a quality measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.

These solutions are then compared through the computation of different quality indices (i.e., *Silhouette-based indices*) to measure the cohesion and separation of each cluster set. The top three configurations, which identify a good partition of the original collection, are selected. ESCAPE includes two variations of the standard Silhouette index to evaluate the quality of the discovered cluster set: (i) the *purified silhouette* index (PS) [5], (ii) the *weighted purified distribution of the silhouette* index (WS) [5], the *average silhouette index* (ASI) [11] and (iv) the *global silhouette index* (GSI) [11]. For these indicators, higher values correspond to better clustering validity. A detailed description of all the computation of these metrics is reported in Section 3.4.

We apply a rank function for each quality index used to quantify the goodness of each partition at the variation of the number of clusters. These solutions are compared through the computation of the three different Silhouette-based quality indices defined before (i.e., Average Silhouette Index, Global Silhouette Index, Weighted Silhouette). These indices are used to measure the cohesion and separation of each cluster set. Firstly, we define a rank from 2 to the maximum number of clusters set by the analyst during the analytics phase, one rank for each index separately. Then, a global score function is defined as follow:

$$Score = (1 - rank\_GSI/K_{max}) + (1 - rank\_ASI/K_{max}) + (1 - rank\_WS/K_{max})),$$

where $K_{max}$ is the maximum value of clusters, while *rank_GSI*, *rank_ASI* and *rank_WS* are the ranks of the Average Silhouette Index, Global Silhouette Index and Weighted Silhouette, respectively. The score lies in the range $[0, (3 - \frac{3}{K_{max}})]$. The worst case is when all the ranks are the smallest for a particular K value, while the highest one is when all the ranks are 1. Lastly, a final rank sorts all these scores. ESCAPE selects the best value for each experiment. In Table 3.2, an example is reported. We reported in bold the best configuration found. We also included a

| Number of Clusters | GSI | ASI | Weighted - Silhouette | rank_GSI | rank_ASI | rank_WS | Score | Rank-Solution |
|---|---|---|---|---|---|---|---|---|
| 2 | 0.210 | 0.239 | 0.290 | 19 | 18 | 18 | 0.105 | 18 |
| 3 | 0.294 | 0.244 | 0.296 | 16 | 17 | 17 | 0.368 | 17 |
| 4 | 0.255 | 0.237 | 0.290 | 18 | 19 | 19 | 0.053 | 19 |
| 5 | 0.332 | 0.315 | 0.370 | 9 | 4 | 4 | 2.105 | 4 |
| 6 | 0.307 | 0.256 | 0.309 | 14 | 16 | 16 | 0.579 | 16 |
| 7 | 0.383 | 0.354 | 0.405 | 1 | 2 | 2 | 2.737 | 2 |
| 8 | 0.345 | 0.315 | 0.365 | 4 | 5 | 6 | 2.211 | 3 |
| 9 | 0.329 | 0.301 | 0.352 | 11 | 11 | 11 | 1.263 | 11 |
| **10** | **0.383** | **0.357** | **0.409** | **2** | **1** | **1** | **2.789** | **1** |
| 11 | 0.290 | 0.295 | 0.347 | 17 | 12 | 12 | 0.842 | 14 |
| 12 | 0.340 | 0.312 | 0.365 | 5 | 7 | 5 | 2.105 | 4 |
| 13 | 0.336 | 0.306 | 0.358 | 7 | 10 | 10 | 1.579 | 9 |
| 14 | 0.320 | 0.322 | 0.376 | 13 | 3 | 3 | 2.000 | 6 |
| 15 | 0.333 | 0.314 | 0.364 | 8 | 6 | 7 | 1.895 | 7 |
| 16 | 0.336 | 0.311 | 0.363 | 6 | 8 | 9 | 1.789 | 8 |
| 17 | 0.322 | 0.311 | 0.364 | 12 | 9 | 8 | 1.474 | 10 |
| 18 | 0.371 | 0.281 | 0.336 | 3 | 15 | 15 | 1.263 | 11 |
| 19 | 0.330 | 0.284 | 0.337 | 10 | 14 | 14 | 1.000 | 13 |
| 20 | 0.306 | 0.285 | 0.338 | 15 | 13 | 13 | 0.842 | 15 |

Table 3.2 Rank function example for a dataset.

plot of the indices' values for each number of clusters in Figure 3.2. In ESCAPE, the analyst can choose how to set the value of the number of clusters through the setting of a parameter. However, our framework proposes as the maximum value for analysis (a default configuration), the average document length for each corpus. In fact, we hypothesise that every word in the document belongs at most to a different topic. In this way, we set an upper-bound for the value of the number of clusters. However, if the average document length is greater than the number of documents in the corpus under analysis, then the value is set to the average frequency of the term. However, these choices can be changed by each analyst, since the framework being distributed is able to analyse several solutions in parallel.

### 3.3.2 Probabilistic model

A completely different approach from the one presented in the previous Section, is the probabilistic topics modelling approach. This technique represents textual documents as probabilities of words and aims to discover and annotate large archives of texts with thematic information. Probabilistic topic modelling algorithms are based on statistical methods that analyse the original texts and their words in order to discover the arguments they go through, and to which other documents they are related. These algorithms are able to describe corpora of documents without previous knowledge of the datasets. In ESCAPE the Latent Dirichlet Allocation (LDA) is integrated, enriched with a self-tuning strategy. LDA is one the most famous and

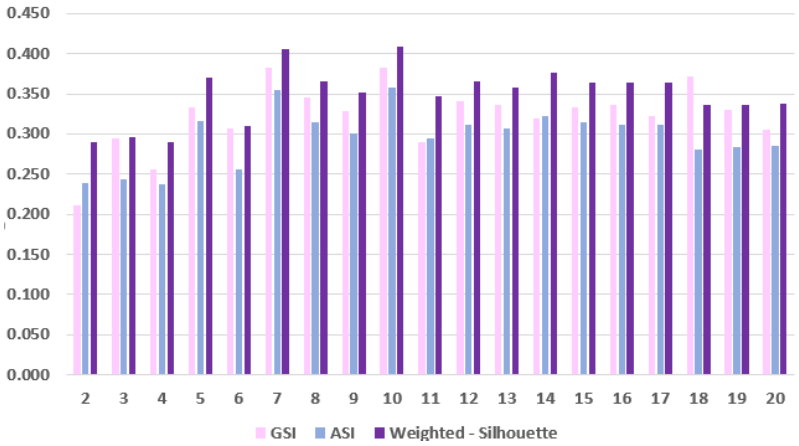



Fig. 3.2 Plot of the silhouette-based indices.

most used probabilistic topic modelling algorithm. The intuition behind LDA is that documents are mixtures of multiple topics [29]. Topics are defined to be distributions over a fixed vocabulary. Documents, instead, are seen as a distribution over the set of different topics, thus showing multiple topics in different proportions. Finally, the LDA algorithm models the given textual dataset with a document-topics and a topic-terms probabilities distribution. LDA can be used to infer the topic hidden in a textual dataset. However, as most of the topic modelling algorithms, LDA requires the number to topics to be previously known and defined. However, finding the optimal number of topic value that have to be discovered using the LDA is not trivial, and it is a open research issue in the scientific community [4].

### Latent Dirichlet Allocation

The Latent Dirichlet Allocation (LDA) is a generative probabilistic model for collections of discrete data such as text corpora [42]. LDA is a Bayesian method for topic extraction in a collection of documents. The goal of topic modelling is to automatically discover the topics from a collection of textual data.

Using Bayesian inference (posterior inference), LDA infers the hidden structure to discover topics inside the collection under analysis. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterised by a distribution over words. LDA is a mathematical method for estimating both of these at the same time: finding the mixture of words that is associated with

each topic, as also determining the mixture of topics that describes each document. It treats each document as a mixture of topics, and each topic as a mixture of words. A topic is formally defined as a distribution over a fixed dictionary. However, these topics are specified a-priori (technically, the LDA model assumes that the topics are generated first, before the documents). Then, for each document in the collection, words are generated through a two-stage process:

1. A distribution over topic is randomly chosen.
2. For each word in the document:
    a) a *topic* is randomly chosen from the distribution defined at the previous step (Step 1).
    b) a *word* is randomly chosen from the corresponding distribution over the dictionary.

This simple intuition highlights that documents present more topics. Each document exhibits topics in different proportions (step 1); then, each word in each document is drawn from one of the topics (step 2b), where the selected topic is chosen from the per-document distribution over topics (step 2a). Therefore, all the documents in the corpus share the same set of topics, while the proportions of these topics in which each topic is exhibited are different.

The LDA modelling describes topics and words as probabilistic distributions from which the document terms will be drawn. Documents are then seen as a distribution over a mixture of latent topics, since each term of a document is drawn from the vocabulary taking into account the terms' probabilities for each given topic of the document's mixture [29]. Specifically, to generate each document in the corpus, the steps performed are:

1. Choose the number of terms from a Poisson distribution;
2. For each of the document's words:
    - Choose a topic $z_n$ from Multinomial($\boldsymbol{\theta}$), where $\boldsymbol{\theta}$ is a Dirichlet($\boldsymbol{\alpha}$), representing the document-topics distribution;
    - Choose a word $w_n$ from Multinomial($\boldsymbol{\phi}_{z_n}$), where $\boldsymbol{\phi}$ represents the topic-words distribution ($\boldsymbol{\phi} \sim$ Dirichlet($\boldsymbol{\beta}$)), conditioned on the topic $z_n$.

Hence, the joint multivariate distribution for the whole corpus of the document-topics distribution $\boldsymbol{\theta}$, the set of $K$ topics $\mathbf{z}$, and the set of $N$ terms $\mathbf{w}$ are defined as:

$$p(\mathscr{D}|\boldsymbol{\alpha},\boldsymbol{\beta}) = \prod_{d=1}^{K} \int p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \left( \prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_n|\boldsymbol{\theta}_d) p(w_{dn}|z_{dn},\boldsymbol{\beta}) \right) d\boldsymbol{\theta}_{\mathrm{d}},$$

where

- $\boldsymbol{\alpha}$ represents the concentration for the prior placed on documents' distributions over topics ($\theta$). This means that low $\alpha$ values will create documents that likely contain a mixture of only few topics, while high values will place more weight on having documents composed of many dominant topics.
- $\boldsymbol{\beta}$ describes the concentration for the prior placed on topics' distributions over terms. This means that low $\beta$ values will likely produce topics that are well described just by few words, while high values will create topics composed by a mixture of most of the words (and so not any word specifically).

With a corpus of documents $X$ in input, the generative LDA model can then be used to support inference of the posterior distribution of the latent variables for the given corpus. Generally, computing these distributions it is unfeasible, and thus it is impossible to exactly solve this posterior Bayesian inferential problem. To overcome this problem, several approximate inference algorithms have been proposed in literature: the ESCAPE engine exploits the Online Variational Bayes algorithms [133], while $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are set to maximise the log likelihood of the data under analysis.

### Self-tuning LDA

The number of topics $K$ in the LDA model, is one of the main goals of this dissertation. As a matter of fact, the more accurate the number of topics given to the model, the better are the clustering results given by the probabilistic model. In literature, different solutions in order to find the most suitable $K$ have been explored and proposed.

As reported in [117], the research has not yielded to easy way to choose proper values for $K$ beyond a major iterative approach. The proposed approach is still iterative,

as all the approaches known so far in literature: this means that in the framework, several LDA models with different values for the $K$ parameter will be created. The goal of the research is to find the optimal K values evaluating not only probabilistic quality metrics but base the evaluation of the models on the topic content. A trade-off between the computational costs and the goodness of the results will be considered, to make the index efficient and effectively usable, even when applied to large data volumes. The newly proposed approach, called ToPIC-Similarity [4], is described in detail in the following paragraph, while a pseudocode of the implemented and proposed algorithm is reported in Algorithm 3.

### Topic-Similarity index

In order to identify a suitable number of topics (i.e. the desired number of clusters) to divide the corpus into, ESCAPE uses a novel proposed strategy named Topic Similarity to assess how topics are semantically diverse and choose proper configurations for the LDA modelling. Given a lower and an upper bound number of clusters set by the analyst (i.e, $[K_{min}, K_{max}]$) a new LDA model is generated for each $K$ value. For each of these partitions, ToPIC-Similarity requires three steps to be gone through [4]:

1. *topic characterisation*, to describe each $t$ topic ($0 \leq t < K$) with the most $n$ representative words;
2. *similarity computation*, to assess how the topics in the same partitioning are similar;
3. *K identification*, to find good clustering configurations to be proposed to the analyst;

Steps 1) and 2) are repeated for all the $t$ topics in every $K$ clustering model.

**Topic characterisation.** To determine the Topic-Similarity, each topic $t$ has to be described with its $n$ most representative terms. This number of words is automatically set, depending on:

- variety of the corpus dictionary $|V|$,

- the average frequency of the terms in the corpus (*AvgFreq*),
- the Type-Token Ratio (*TTR*, $\in [0, 1]$) and the currently considered number of topics *K*.

To select the number of representative words, ESCAPE considers only the richest part of the corpus under analysis (by means of the TTR index, which represents the lexical variation of the corpus) and then samples the remaining words by the average frequency of the terms. This quantity, named $Q$, represents the total number of considered terms, and it is mathematically defined as: $Q := \frac{|V| \cdot TTR}{AvgFreq}$. Given $Q$, the number of words $n$ describing a single topic $t$ is given by:

$$n = \begin{cases} \frac{Q}{K} & if\ Q \geq K \cdot AvgFreq \\ AvgFreq & if\ Q < K \cdot AvgFreq \end{cases} \tag{3.2}$$

For each topic $t$, the number of $n$ terms is obtained taking the corpus dictionary sorted by the distribution $\boldsymbol{\phi}$.
This is done if the final number of words considered for each topic is greater than the average frequency of terms of the corpus, to make the sampling reasonable (it does not make sense to sample with a period greater than the number of items themselves) and to have every topic represented at least by a number of words equal to the average frequency. If the condition is not satisfied, then a minimum number of terms is considered. We set this lower bound quantity to be equal to the average frequency of the terms in the corpus. Repetitions among the considered terms characterising each topic are removed and the resulting words are considered together to make the topic representations comparable. Then, for each term in every topic, if the word is present in the topic description, the correspondent value is set to the probability that the term has to be picked up in the topic, or to 0 if it is not.

**Similarity computation.** To compute the ToPIC-Similarity index of the partitioning, all the possible pairs of topics are considered. Similarity among all the topics is computed through the *cosine similarity*. Cosine similarity is one of the most used techniques in information retrieval and data mining, especially in text analysis and topic modelling, because of its efficiency [134] and ability to reflect the human perception of similarity [135]. It is derived from the Euclidean dot product and, given two topics *t'* and *t"* of the same partitioning *K*, it is computed as follows:

$similarity(t',t'') = \frac{\mathbf{N}_{t'} \cdot \mathbf{N}_{t''}}{\|\mathbf{N}_{t'}\|_2 \|\mathbf{N}_{t''}\|_2}$, where $\mathbf{N}_{t'}$ and $\mathbf{N}_{t''}$ are the set of representative words of topic *t'* and *t''*, respectively.
The result is a $K \times K$ symmetric matrix where each cell $(i,j)$ is the similarity between topic in row i and topic in column j.
Since we use the cosine similarity in the text analysis context and the probabilities of the terms are always positive, the obtained values will always be in the interval from 0 to 1. The ToPIC-Similarity index for the considered clustering is obtained averaging the similarity matrix over $K$ to keep in consideration the different number of clusters. For this issue, the *norm* of the whole similarity matrix (using the Frobenius norm) is computed and then the obtained values are divided by $K$. Since ToPIC-Similarity is expressed in percentage, the previous values are multiplied by 100.

**K identification.** A topic similarity function is obtained, computing ToPIC-Similarity for the several LDA models generated for different $K$. To find optimal $K$ values a trade-off approach between optimal results and computational cost has been chosen. It has been empirically seen that the obtained Topic Similarity function is, in most cases, decreasing but not monotonic. Two conditions are defined to choose as $K$ values:

1. the local minima of the curve, namely the $K$ for which $ToPIC\text{-}Similarity(K_i) < ToPIC\text{-}Similarity(K_{i+1})$;
2. the only points belonging to a decreasing segment of the curve. Thus, the second derivative is computed and only the points that have a positive second derivative are considered.

In our study, we considered the selected values to be the first three points that satisfy both of the above conditions. The topic modelling and the search for optimal $K$ values can stop when the first three values are found, or when the algorithm reaches the $K$ upper bound value set by the analyst (and in this case a lower number of optimal values will be proposed to the analyst).

For each strategy implemented in ESCAPE, the optimal three values of K (i.e., the number of topics) are reported to the analyst, as possible good partitions.

**Algorithm 3:** ToPIC-similarity pseudo-code

**Input :** $X$, $K_{min}$, $K_{max}$
**Output :** *kSol*

```
// variable inisialisation
topicS = [ ],   NTerms = [ ];
for K ← K_min to K_max do
    // build the LDA model;
    LDAModel ← lda.fit(X);
    Q ← (|V| · TTR)/AvgFreq;
    // set the number of terms per topic;
    if Q ≥ K · AvgFreq then
        n ← Q/K);
    else
        n ← AvgFreq;
    end
    // collect together the terms of each topic;
    for t ← 0 to (K-1) do
        NTerms.append(LDAModel.describeTopics()[t].sort().take(n));
    end
    N ← NTerms.size();
    topicsDescr = zeros(K, N);
    simMatrix = zeros(K, K);
    for t ← 0 to (K-1) do
        for word ← 0 to N do
            // take the probability that the term has to be drawn
            // from the topic, given the LDAModel
            topicsDescr[t][word] ← LDAModel.describeTopics()[t,
              NTerms[word]];
        end
    end
    for t ← 0 to (K-1) do
        for s ← 0 to (K-1) do
            simMatrix[t][s] ← cosine(topicsDescr[t], topicsDescr[s]);
        end
    end
    topicS.append(Frobenius-norm(simMatrix)*100/K);
    if topicS[K] ≥ topicS[K-1] AND secondDerivative(topicS[K-1]) > 0 then
        kSol.append(topicS[K-1]);
        if kSol.size() > 3 then
            return kSol.take(3);
        end
    end
end
```

### 3.3.3 Complexity of algorithms

The complexity of the algorithms exploited for both the approaches is reported in the following subsections.

#### Joint-approach

As reported in the previous Section, K-Means is one of the most commonly used clustering algorithms that clusters the data points into a predefined number of clusters. In ESCAPE, we integrate a parallelised variant of the *K-Means++* method, called *K-Means||* [136], presented in *spark.mllib* library [137].

The advantage of K-Means is its simplicity: starting with a set of randomly chosen initial centers, one repeatedly assigns each input point to its nearest center, and then recomputes the centers given the point assignment. This local search, called Lloyd's iteration, continues until the solution does not change between two consecutive rounds.

From a theoretical standpoint, K-Means is not a good clustering algorithm in terms of efficiency or quality: the running time can be exponential in the worst case [138, 139] and even though the final solution is locally optimal, it can be very far away from the global optimum (even under repeated random initialisations). Nevertheless, in practice the speed and simplicity of K-Means cannot be beat. Therefore, recent work has focused on improving the initialisation procedure: deciding on a better way to initialise the clustering dramatically changes the performance of the Lloyd's iteration, both in terms of quality and convergence properties [140].

The complexity of K-Means|| is discussed in [136] in which the authors demonstrate its practical effectiveness. The running time of K-Means|| consists of two components: the time required to generate the initial solution and the running time of Lloyd's iteration to convergence [136]. Let $k$ be the number of topics and $d$ the number of documents belonging to a corpus. Their main idea is that instead of sampling a single point in each pass of the K-Means++ algorithm, they sample O(k) points in each round and repeat the process for approximately O(logd) rounds. At the end of the algorithm, they obtain O(klogd) points that form a solution that is within a constant factor away from the optimum. Then the authors recluster these

O(klogd) points into k initial centers for the Lloyd's iteration. K-Means|| is quite simple and lends itself to easy parallel implementations.

However, this algorithm analysis turns out to be highly non-trivial, requiring new insights, and is quite different from the analysis of K-Means++. The performances of this algorithms have been evaluated on real-world datasets. The main observations in the experiments are:

- O(logn) iterations is not necessary and after as little as five rounds, the solution of K-Means|| is consistently as good or better than that found by other methods;
- The parallel implementation of K-Means|| is much faster than existing parallel algorithms for K-Means;
- The number of iterations until Lloyd's algorithm converges is smallest when the seed is set using K-Means||.

Distributed version of K-Means is roughly O(k); however, when the number of cluster increases, a slower start is expected. In this case, a variant of Lloyds is used which is roughly O(ikdt), where $i$ is the maximum number of iterations to run, $k$ is the number of desired clusters, $d$ is the number of documents, and $t$ is the number of distinct terms [136].

### Probabilistic approach

To solve the big topic modelling problem, both time and space complexities of batch Latent Dirichlet Allocation (LDA) algorithms need to be reduced. Although parallel LDA algorithms on the multi-processor architecture have low time and space complexities, their communication costs among processors often scale linearly with the vocabulary size and the number of topics, leading to a serious scalability problem. LDA supports different inference algorithms via *setOptimizer* function. *EMLDAOptimizer* learns clustering using expectation-maximisation on the likelihood function and yields comprehensive results, while *OnlineLDAOptimiser* uses iterative mini-batch sampling for online variational inference and is generally memory friendly.

In [141], the complexity of the inference in LDA is analysed. First, the author study the problem of finding the maximum a posteriori (MAP) assignment of topics to

words, where the document's topic distribution is integrated out. We show that, when the effective number of topics per document is small, exact inference takes polynomial time. In contrast, the authors show that, when a document has a large number of topics, finding the MAP assignment of topics to words in LDA is NP-hard. There are many potential applications of MAP inference of the document's topic distribution. For example, the distribution may be used for topic-based information retrieval or as the feature vector for classification. Moreover, batch EM has high time and space complexities to learn big LDA models from big data streams [142]. This algorithm has constant memory requirements, since it requires full pass through the entire corpus each iteration. Therefore, it is not naturally suited when new data is constantly arriving.

To this aim, authors in [133] propose a scalable online Variational Bayes (VB) algorithm for Latent Dirichlet Allocation (LDA). Online LDA is based on online stochastic optimisation with a natural gradient step, which converges to a local optimum of the VB objective function. The authors demonstrate that online LDA finds topic models as good or better than those found with batch VB, and in a fraction of the time [133]. Its implementation only looks at a subset of the total corpus of documents each iteration, and thereby is able to find a locally optimal setting of the variational posterior over the topics more quickly than a batch VB algorithm could for large corpora. In *spark.mllib* library, both implementations are available, while in ESCAPE the *OnlineLDAOptimiser* has been integrated.

## 3.4 Knowledge validation and visualisation

Evaluating data models using unlabelled data is a complex and time-consuming task. Many theoretical quantitative indices can be used to assess the quality of a clustering process and so identify the best partitioning. However, it is good practice to verify if the obtained models actually satisfy the main expectation. To evaluate the goodness of ESCAPE in discovering good topic partitions, and confirm the ranking obtained with the quality metrics, different visualisation methods can be used. This is possible because the document clustering and topic detection results allow to directly visualise the clustering inferred in the learning process, both in terms of documents grouping and in terms of topics characterisation. Evaluating the quality of a set of clusters obtained on high dimensional datasets could require several strategies to

visualise different aspects of the data under analysis, such as indices aggregating the many dimensions of the data items. This section presents several interesting and innovative validation (both quantitative approaches and visualisation techniques), integrated in ESCAPE to visualise and validate the document partitions discovered on a given collection of textual documents.

The aim of this component is to visualise and make the information and the extracted knowledge easy to be interpreted at different levels of detail. To this extent, ESCAPE includes interactive and navigable dashboards tailored to different stakeholders, providing both domain specific information and high-level overviews. Indeed, the dashboards can be customised for each end-user, providing deep targeted knowledge for domain experts and human-readable informative contents for non-expert users.

Besides displaying only statistical values or technical diagrams, which are often difficult to interpret, ESCAPE proposes several plots to explore and visualise the knowledge extracted from textual corpora.

### 3.4.1 Model analysis and validation

Different visualisation techniques help the end-user to better understand data under analysis and the corresponding discovered knowledge. In ESCAPE, several visualisation techniques have been integrated to represent several interesting facets at different granularity levels for highlighting significant results. In this way, depending on the end-user, the data are translated into graphs and charts, making easily their exploration.

To this aim, we proposed two types of dashboard:

- **Technical dashboards**: which are designed to create reports for the domain expert which synthesise data from multiple sources to streamline reporting processes. With their exploitation, analysts are able to understand how the algorithms work and to analyse the parameter setting of each algorithm in each text mining phase.
- **Informative dashboards**: which include several graphical representations that are self-explained. These proposed graphical representations are exploited

to simplify and synthesise the extracted knowledge patterns in a compact, human-readable, detailed and exhaustive representation.

These types of dashboard are used to display information tailored to different specific users, including specific visualisation for the different stakeholders, and not considering a single user's perspective.

### 3.4.2 Quantitative validation

To quantitatively measure the performance of the algorithms, a set of qualitative metrics were included. Starting from well-known metrics used in literature, ESCAPE develops new quantitative methods for the validation, which are described in the next subsection. Specifically, for the join-approach we integrated the silhouette-based indices, while for the probabilistic model two metrics are explored: (i) the perplexity and (ii) the entropy.

#### Silhouette-based indices

To analyse the clustering outcomes two indicators based on the calculation of the widely used silhouette [132] are proposed. The silhouette index is a quality measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.

In particular, let us consider a single document $i$ and the cluster to which the document has been allocated. For each document $i$, let $a_i$ be the average distance between $i$ and the other documents in the same cluster. $a_i$ can be interpreted as a measure of how well $i$ is assigned to its cluster (the smaller the value, the better the assignment). Let $b_i$ be the lowest average distance between the document $i$ and each one of the other clusters (not containing the document $i$).

The silhouette is defined as:

$$s_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

These solutions are then compared through the computation of different quality indices (i.e., Silhouette-based indices) to measure the cohesion and separation of

each cluster set. The top three configurations, which identify a good partition of the original collection, are selected. ESCAPE includes two variations of the standard Silhouette index to evaluate the quality of the discovered cluster set: (i) the purified silhouette index (PS), (ii) the weighted purified distribution of the silhouette index (WS), the average silhouette index (ASI) and (iv) the global silhouette index (GSI). The PS index [5] disregards documents that appear in singleton clusters. Thus, the impact of these documents in the overall Silhouette index is reduced, while the WS index (assuming values in [0; 1]) [5] represents the percentage of documents in each positive bin properly weighted with an integer value $w \in [1;10]$ (the highest weight is associated with the first bin [1-0.9], and so on) and normalised within the sum of all the weights. The higher the weighted silhouette index, the better the identified partition. Moreover, distributions with a positive asymmetry (i.e., many documents with silhouette values in the higher bins) are preferred to those with a predominance of lower silhouette values (negative skewness).

The other two indicators (i.e., ASI and GSI) are based on the previous definition of Silhouette, whose expressions contain the set $C_k$ of the patterns belonging to cluster $k = 1,...,K$; the cardinality $|C_k|$ of cluster $C_k$ (load patterns $i$ belonging to the cluster $C_k$), and $N$ the total number of load patterns clustered (i.e., the number of consumers).

The average silhouette index (ASI) [11] is expressed as

$$ASI = \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in C_k} s_i,$$

while the global silhouette index (GSI) [11] is expressed as

$$GSI = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i \in C_k} s_i.$$

For these indicators, higher values correspond to better clustering validity. While ASI gives an overview of the average silhouette of the entire cluster set, GSI is able to take into account the possible imbalance number of elements in each cluster. Clusters with large number of documents are penalised in the GSI computation. The higher the values, the better the clustering partition. ESCAPE automatically rank the solutions, according these three indicators, and plot them using a bar chart. An

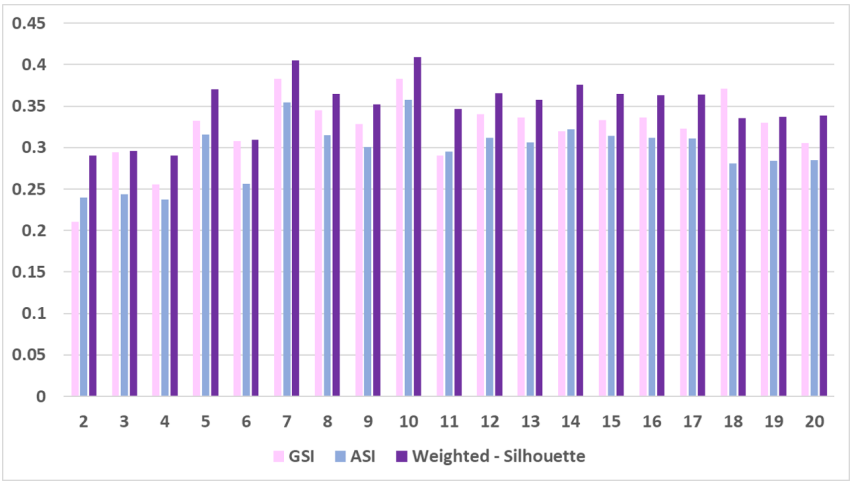



Fig. 3.3 Example of bar-chart representation for the analysis of the silhouette-based indices.

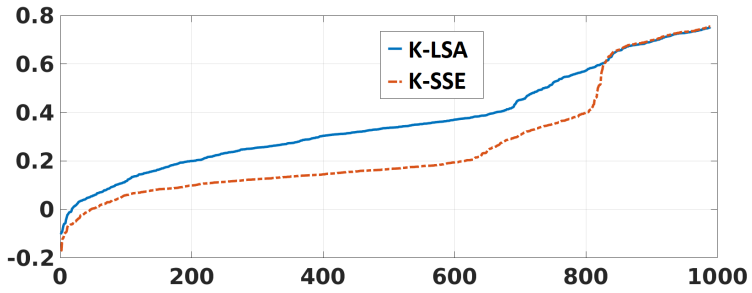



Fig. 3.4 Comparison of the ordered distribution of the purified-silhouette of two different partitions obtained by ESCAPE.

example of this representation is reported in Figure 3.3. A rank function is defined to reported to the analyst only the best solution for the experimental sets. Moreover, ESCAPE plots the ordered distribution (i.e., duration curve representation [11, 12]) of the purified silhouette, which is useful to compare different partitions of the same dataset. A duration curve illustrates the variation of the purified silhouette such that the smallest value is plotted in the left and the greatest one in the right. An example is reported in Figure 3.4.

### Perplexity

The perplexity is a measure of the quality of probabilistic models, that describes how well a model predicts a sample (i.e. how much it is perplexed by a sample from the

observed data). It is frequently used to assess the performance of language models and to evaluate LDA models in the context of document modelling [42]. The authors of the LDA model used it to evaluate and compare the results of the LDA topics inference. Perplexity is monotonic decreasing in the likelihood of the data and is equivalent to the inverse of the per-word likelihood. It is defined as:

$$Perplexity(D) = \exp\left\{ - \frac{\sum_{d=1}^{D} \log p(w_d)}{\sum_{d=1}^{D} N_d} \right\}$$

Here $D$ is the number of documents (the corpus under analysis), $w_d$ represents the words in document $d$, $N_d$ the number of words in document $d$. Given a calculated model, the lower the general perplexity, the better the model performance and the probability estimate of the corpus [143].

### Entropy

In information theory, entropy [144] is defined as the amount of information contained in a transmitted message (i.e., the event of interest). Generally, the greater the uncertainty in an event, the more information it will contain. This means that information decreases in uncertainty or entropy. With this definition, entropy can be seen as an average ambiguity of a probabilistic event: the greater is entropy, the greater the uncertainty and ambiguity of the event. Applied to the modelling context, entropy measures how uncertain the model is: the lower the entropy of the model, the more certain it is that the model is describing the corpus under analysis. Specifically, for each $d$ document in the corpus $D$ we calculated that entropy must belong to one of $K$' s topics and it is calculated as follows:

$$H(d) = -\sum_{k=1}^{K} p(d=k)\log(p(d=k))$$

where $p(d=k)$ is the probability that the considered document will be assigned to topic $k$. To compute the entropy of the whole clustering model, we averaged the entropy of each document on the whole corpus: $H(model) = \frac{\sum_{d=1}^{D} H(d)}{D}$.

### 3.4.3 Visualisation techniques

Visualising the result of clustering algorithms is not a trivial task for high-dimensional data such as textual collections. Considering only the cardinality of documents labelled in the same clusters is not sufficient to describe the clustering results; for this reason, in this dissertation several visualisations techniques have been included and proposed to show interesting correlation among data under analysis. These visualisations are able to show different information related to the topic-term and document-topic distributions at different granularity levels.

#### t-SNE

*t-Distributed Stochastic Neighbor Embedding* (t-SNE) [145] is an innovative visualisation technique to represent high-dimensional data into a two or three dimensional map, suitable for human observation. The technique is a variation of Stochastic Neighbour Embedding proposed in [146], based on a non-linear dimensionality reduction algorithm. While the SNE minimises the sum of Kullback-Leibler divergences [147] overall data points using a gradient descent method to measure the minimisation of sum of difference of conditional probability, t-SNE minimises the sum of the difference in conditional probabilities using a symmetric version of the SNE cost function, with simple gradients.

Normally, the high-dimensional dataset is represented graphically by reducing the dimensionality of the dataset, while attempting to preserve the most significant data structure. This approach is usually achieved with reduction techniques such as Principal Component Analysis (PCA) [115]. However, especially when the high-dimensional dataset consists of similar data points, it is often difficult to use a linear mapping (as the one performed by the Principal Component Analysis) to correctly display the differences between the data elements. As a matter of fact, t-SNE outputs provide better results than PCA and other linear dimensionality reduction models. This is because a linear method such as classical scaling is not good at modelling curved manifolds [145]. It focuses on preserving the distances between widely separated data points rather than on preserving the distances between nearby data points.

This feature allows to represent similar data points close to each other and, in the meanwhile, different data points to be represented far in the low-dimensional

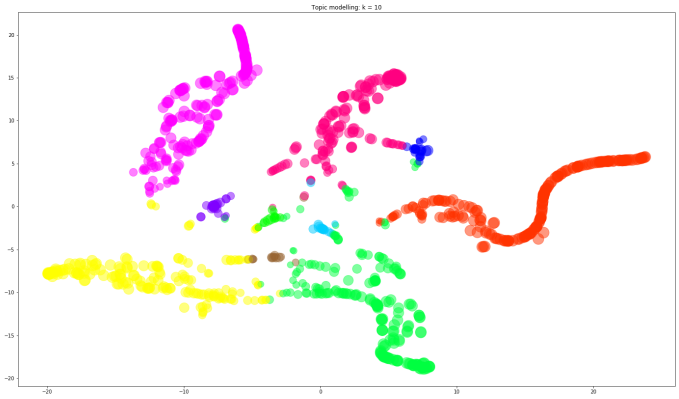



Fig. 3.5 Example of the t-SNE representation.

space. The algorithm converts the Euclidean distances among the data points into conditional probabilities representing similarities. $P(x_i|x_j)$ is the probability that a certain point $x_i$ would pick $x_j$ as its neighbour, if neighbours were distributed with a Gaussian probability centred in $x_i$. This probability should be higher for nearby data points, while it will be almost zero for very different data points. The same probabilities are also computed for the new low-dimensional space. In this case, $P(y_i|y_j)$ is the probability that the low-dimensional data point $y_i$ would choose $y_j$ as its neighbour. Trivially, if the two spaces have the same dimensions (i.e., there is no reduction), then the two conditional probabilities are the same. Data reduced in dimensionality can be printed, while being able to display the original structure and the relationships between the data points. The colouring of the points reflects the assignment to a specific topic (i.e., cluster) after the two integrated methodologies. An example of t-SNE representation is reported in Figure 3.5.

### Topics-terms representations - termite

To analyse and characterise the topic distributions obtained from the clustering and topic moddeling, the topic description distribution is analysed. In order to support an effective evaluation of the term distributions associated with the discovered topics, authors in [148] proposed a new visualisation technique, named *termite*. This kind of representation allows the analyst to evaluate the quality of the clustering results, considering the goodness of each topic but also the entire clusters.

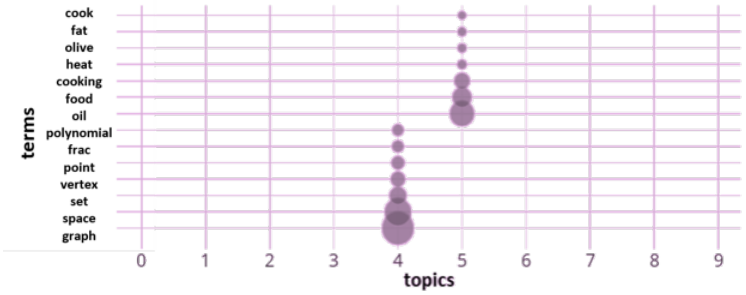



Fig. 3.6 Example of the termite representation.

For each topic, the most representative words (terms) are taken: the belonging of each term is represented as a point in the plot, whose size depend on the probability that that term should be taken from the topic during the creation of the document. By this way, the topic-term distribution helps the analyst to detect salient terms for single topics, or if a word is equally distributed among the various topics during the document creation process. This plot is also useful to select a threshold to remove words during the pre-processing phase. An example of termite representation is reported in Figure 3.6.

### Word clouds

A word cloud [149] is a popular visualisation of words typically associated with textual data. They are most commonly used to highlight salient or relevant terms based on frequency or probability in a collection. A word cloud is a beautiful, informative image that communicates much in a single glance. Selecting a threshold or a maximum number of words to be drawn from a topic, the analyst is able to analyse several clusters jointly. This format is useful for quickly perceiving the most prominent terms. In this thesis, word clouds have been created utilising the words used to describe each cluster content obtained by the clustering and topic modelling [150].

The clouds represent the topic-term distribution: the comparison of the clouds obtained is left to the human analyst judgement. The word cloud display emphasises the terms with the highest probabilities with a larger font size. In this way it is possible to observe directly if the results of the clustering are good or if the modelling of the argument has not given acceptable results. For its clarity and simplicity, the representation of word cloud has already been used in the literature to visualise


Fig. 3.7 Examples of the word cloud representation.

the results of the topic discovering [117]. Examples of word clouds are reported in Figure 4.25. Since in most cases a probability distribution is not available, it is possible to transform the frequency distribution of word-counts into frequency distributions. Thus, the extraction of items which have a length of one, could be used to compute the word-cloud representation in case of models that do not return a probability distribution.

### Graph representation

Graphs are convenient widespread data structures used to represent many real-life applications [151, 152], ranking from social networks as proposed in [153] to information retrieval [154], Graph clustering [155], but also textual data to generate a semantic network [156].

By the exploitation of graphs-based techniques, it is possible to provide to the end-user, a different perspective on a process [157]. By this way, only the most relevant subprocesses (i.e., subgraphs) are displayed, rather than the complete, which often results very chaotic in unstructured domains.

An **undirected graph** $G = (V, E)$ is a data structure [158] that consists of following two components:

(i) a finite set $V$ of *vertices* also called *nodes*,

(ii) a finite set $E$ of *unordered pair* of the form $(u, v)$ called *edge*.

The pair of the form $(u, v)$ indicates that there is an edge from vertex $u$ to vertex $v$. The edges may contain weight/value/cost.

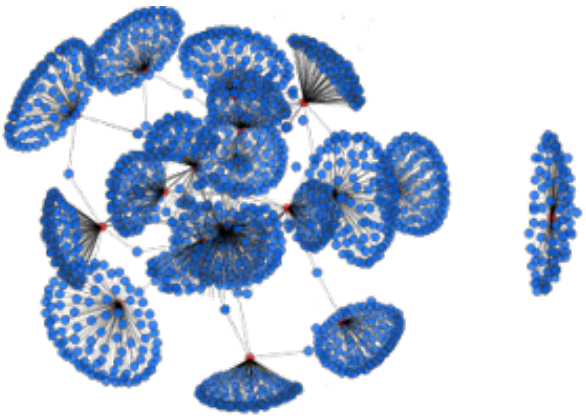


Fig. 3.8 Example of the graph representation.

In this thesis, we propose to use the graph representation to analyse the topic-term distribution, which is reported in Figure 3.8. In this case, we introduce two types of nodes: topic nodes, which are green nodes one for each topic, and term nodes, which are pink nodes one for each distinct term in the collection under analysis. Then, for each topic we add an edge for each term linked with that topic. We fix a threshold to avoid link with low probability. If a words appear in more than one topic, we colour the node in red. By this way, we are able to compute the connectivity of the graph to analyse the results of the clustering and topic modelling. A graph is said to be connected if every pair of vertices in the graph is connected [159]. A graph which is not connected is also called disconnected. A graph G is disconnected if there exist two nodes in G such that no path in G has those nodes as endpoints. If a topic is characterised by words that are not used in others topic, that topic will be disconnected by the others. This means that the number of clusters selected by ESCAPE is able to separate the different topics. As a matter of fact, if all the words are connected to each other, all the terms have the same probability of belonging to each cluster.

### Word tables

Another way to represent the topics and their content is to use word tables. This is a very simple representation for analyse the topic-term distribution, essentially lists the terms that describe the topics in descending order of probability. As suggested in previous works [148], the quality of a topic is often determined by the consistency of its constituent words. The objective of this type of visualisation is to evaluate how

the clustering process has been carried out, considering the arguments of cohesion and coherence through their content. This representation helps the analysis of the graph presented in the section before. Also, in this case, the use of a threshold could help the analyst to exclude rare terms in each topic, and highlight the main salient terms.

This type of results could be also extended including the extraction of association rules able to characterise each cluster content. In this way, we are able to extract the most interesting correlations between words in each cluster.

### Correlation matrix maps

Correlation matrix maps can be used to analyse the possible correlation between different topics as proposed in [5]. Five different coloured correlation ranges have been used: [0.87-1.00] black, [0.75-0.87] dark gray, [0.62-0.75] gray, [0.5-0.62] light gray and 0.0-0.5 white (these bins are symmetrical for negative correlations). Documents are first sorted by topic, and then the dot products between all document pairs are computed. Thus, documents belonging to the same macro category tend to be more similar to each other than those belonging to different ones. Using this simple plot, we are able to analyse possible correlations between different categories. If in the correlation plot are present only dark rectangles in the diagonal of the matrix, it means that there is no longer any interest in reducing the number of topics K found by the algorithms.

An example of this representation is reported in Figure 3.9. When dealing with high dimensional data, analysing correlations could be a problem. To this aim, we also propose to use a graph to visualise correlations. With this technique, as said before, we have a powerful insight on how to model complex processes. Moreover, a labelled graph can be represented with an *Adjacency Matrix*. For the simple case, if we have a zero between column $x$ a row $y$, we know that no edge goes from node $x$ to node $y$. Instead, if we have a 1, we know that an edge goes from node $x$ to node $y$. The correlation matrix is a square matrix with values going from -1 to 1. ESCAPE transforms this matrix into an adjacency matrix. Moreover, since the correlation matrix is symmetric, then the graph have to be undirected.

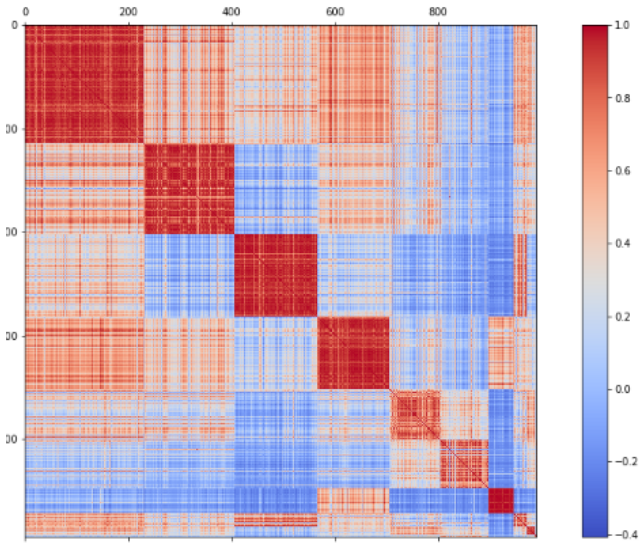



Fig. 3.9 Example of correlation matrix map.

### 3.4.4 Frequent Items

Intuitively, a set of words that appears in several documents, is said to be frequent. To be formal, we assume there is a number, called the support threshold. Given a set of items $I$, the support for $I$ is the number of documents for which $I$ is a subset. It the support of $I$ is grater or equal the fixed threshold, then $I$ is said frequent. Indeed, frequent itemsets are groups of items that often appear together in the data. It is important to know the basics of market-basket analysis for understanding frequent itemsets [160].

The market-basket model of data is used to describe a common form of a *many-to-many relationship* [161] between two kinds of objects. On the one hand, we have items, and on the other we have baskets, also called transactions. The set of items is usually represented in the form of *Attribute* = *Value*. ESCAPE extracts the frequent items from each collection of corpora. ESCAPE transforms each text into a bag-of-word representation, in which is a word appears in the collection, that word is set to *True*, otherwise it is set to *False*. Each text is in this way transformed into a binominal representation. In ESCAPE each transaction consists of a set of words (an itemset). Usually it is assumed that the number of items in a transaction is small, much smaller than the total number of items i.e. in most of the examples several attribute values are 'False'.

The frequent-itemset problem is that of finding sets of items that appear together in at least a threshold ratio of transactions. This threshold is defined by the minimum support criteria. The support of an itemset is the number of times that itemset appears in the dataset divided by the total number of examples. The discovery of frequent itemsets is often viewed as the discovery of association rules, although the latter is a more complex characterisation of data, whose discovery depends fundamentally on the discovery of frequent itemsets. Association rules are derived from the frequent itemsets. In ESCAPE we integrated the FP-Growth algorithm to find the frequent itemsets. These frequent itemsets could be then used to extract the most interesting association rules.

The user can choose to select only the top-k items for each cluster/topic or decided to extract also high-level correlation items through the analysis of the association rules [160].

### 3.4.5 Comparison between different solutions

To compare the different solutions found by ESCAPE, we integrated in ESCAPE the Adjusted Rand Index (ARI) metric. The ARI is the corrected-for-chance version of the Rand index [162], [163] and [164]. The Rand Index [165] is defined as the number of pairs of objects that are either in the same group or in different groups in both partitions divided by the total number of pairs of objects. The Rand Index lies between 0 and 1. When two partitions agree perfectly, the Rand index achieves the maximum value 1. A problem with Rand Index is that the expected value of the Rand index between two random partitions is not a constant. This problem is corrected by the Adjusted Rand index [163] that assumes the generalised hyper-geometric distribution as the model of randomness. The Adjusted Rand index has the maximum value 1, and its expected value is 0 in the case of random clusters. A larger Adjusted Rand index means a higher agreement between two partitions. The Adjusted Rand index is recommended for measuring agreement even when the partitions compared have different numbers of clusters.

Such a correction for chance establishes a baseline by using the expected similarity of all pair-wise comparisons between clusterings specified by a random model. Traditionally, the Rand Index was corrected using the permutation model for clusterings (the number and size of clusters within a clustering are fixed, and all random cluster-

ings are generated by shuffling the elements between the fixed clusters). However, the premises of the permutation model are frequently violated; in many clustering scenarios, either the number of clusters or the size distribution of those clusters vary drastically. For example, consider that in K-means the number of clusters is fixed by the practitioner, but the sizes of those clusters are inferred from the data.

To this aim, ESCAPE reported the ARI between solutions using the same strategy (i.e., joint-approach or probabilistic approach) to compare the different weighting schema impact; but also, the comparison between the two strategies, to analyse which are the main differences between the two approaches.

# Chapter 4

# Experimental results

This Chapter presents the experimental results performed to asses the effectiveness of ESCAPE in discovering well-cohesive and well-separated groups of documents. The datasets and their descriptions are reported in Section 4.1. For each dataset, the main features are computed to provide an overview of each data distribution. The experimental setting is described in Section 4.2.

The analyst could be interested in answering several questions that could arise during the analysis of the different approaches in the corpora.

- How do the different weighting strategies highlight specific words that characterise each corpus? How is it possible to compare the results and the cardinalities of each experiment?
- How the different partitions are similar each others? Which quality metrics could be interesting to analyse?
- Which are the performances with respect to the state-of-the-art methodologies?

Once the partitions have been obtained, the analyst should be analysed the partitions in terms of semantic meaning. Specifically, two types of investigation could be exploit.

- Which is the document-topic distribution in each partition?
- Which is the topic-term distribution in each partition?

- Which are the main differences between the joint approach and the probabilistic approach?

To answer to all these questions, ESCAPE integrates two types of dashboards able to help the analyst during all the phases of the analytics exploration.

The outline of the Chapter is reported below. Section 4.3 and Section 4.4 discuss and report the results obtained for each dataset using the joint-approach and the probabilistic model, respectively. Specifically, for each method integrated in ESCAPE a deeper analysis of the impact of the different weighting schemas is reported, in order to highlight and explain the differences obtained in the gathered results.

Section 4.5 reports in detail the results obtained for a running example, including two exhaustive types of dashboards described in the Chapter 3, including the analysis of the obtained quantitative evaluation indices and the comparison with the state-of-the-art outcomes.

The Chapter ends with Section 4.6, giving a final sum-up and final considerations about the results and the performance of the ESCAPE system.

## 4.1 Experiment datasets

| ID Dataset | Source | Textual data type |
|---|---|---|
| D1 | Wikipedia[a] | Documents |
| D2 | Wikipedia[b] | Documents |
| D3 | Wikipedia[c] | Documents |
| D4 | Twitter[d] | Short messages |
| D5 | PubMed[e] | Articles |
| D6 | PubMed[f] | Abstracts |
| D7 | Reuters[g] | Documents |

Table 4.1 Experiment datasets.

[a] https://en.wikipedia.org/wiki/Wikipedia
[b] https://en.wikipedia.org/wiki/Wikipedia
[c] https://en.wikipedia.org/wiki/Wikipedia
[d] https://crisislex.org/tweet-collections.html
[e] https://www.ncbi.nlm.nih.gov/pubmed/
[f] https://www.ncbi.nlm.nih.gov/pubmed/
[g] http://www.daviddlewis.com/resources/testcollections/reuters21578

The proposed framework has been tested over several datasets, belonging to different domains ranging from social networks and digital libraries (e.g. Twitter, Wikipedia) to scientific papers (e.g. PubMed collections). Corpora have been chosen to have different characteristics, from the number of documents to the length of each individual document, from lexical richness to the average frequency of terms. Moreover in the same corpus, the documents should be characterised by homogeneous lengths and heterogeneous subjects, as well as being produced by different authors. In this way these features allow results to be comparable and generic, avoiding over-fitting of data sets. We group the datasets based on their source and typology, as shown in Table 4.1.

### 4.1.1 Wikipedia

Datasets from D1 to D3 are collected from English documents from the Wikipedia collection[1] which is the largest knowledge-base ever known. However, in spite of its utility, its contents are barely machine-interpretable [166]. Wikipedia's contents are released under *Creative Commons license*, and so their usage is free and public. The Wikipedia project pursues a neutral point of view in its discussion of topics, both in terms of articles that are created and in terms of content, perspective and sources within these articles. The categories of each dataset have been chosen to be sufficiently separate and therefore detectable by the clustering algorithms. For each category, *top-k* articles are extracted, which will form our corpus.

From these categories, different datasets have been generated, divergent by the number of documents extracted for each topic. To construct the first data set (i.e., D1), 200 articles were taken from the following five categories: *cooking*, *literature*, *mathematics*, *music* and *sport*. Instead, the following ten categories were chosen to build the subsequent collections of articles: *astronomy*, *cooking*, *geography*, *history*, *literature*, *mathematics*, *music*, *politics*, *religion* and *sports*. We chose 2500 and 5000 documents, respectively from these ten categories. Table 4.2 shows the *statistical features* of the three Wikipedia data sets used to test ESCAPE. For each dataset, we reported the statistics for both the strategies presented in Chapter 3. In more detail, we analysed each dataset including and excluding the hapax removal block, i.e., words that appear only once in the entire corpus. The idea is that the removal of

[1] https://en.wikipedia.org/wiki/Wikipedia

these words could improve the performance of the algorithms. Table 4.2 shows the proposed indices characterising the data distribution in the Wikipedia datasets with Hapax (WH) and without Hapax (WoH). Indeed, the features with Hapax (WH) and without Hapax (WoH) in Table 4.2 does not show any significant difference. Thus, removing Hapax does not change the main dataset features. By this way, this step could be more significant for the two approaches and affect the results in a different way. As a matter of fact, since the singular value decomposition is not affected by the presence of words that globally appear only once, as their effect is limited in the $k$ components used during the SVD. On the other hand, removing Hapax allows the LDA to construct a more precise probabilistic model. Since each word of the textual collection is drawn from the vocabulary taking into account the probabilities of terms for each given topic of the mixture of documents, removing Hapax words will allow the noise's reduction in the vocabulary.

The data sparsity of each dataset is well described by the TTR index (i.e., Type of Token Ratio), which is able to distinguish between sparse and dense data distribution. High TTR values indicate a high degree of lexical variation, while lower TTR values indicate the opposite. In these three datasets, the TTR index falls into the range [0.01, 0.02] for the WoH and into the range [0.03, 0.04] for the WH. Thus, dataset with the smallest TTR is denser than the other two. However, TTR alone cannot show the complete lexical complexity. Conversely, the Guiraud index is able to describe the lexical richness of a document corpus. The three datasets are characterised by a large number of terms (words with repetition) and by a very large dictionary (number of distinct words without repetition). Each word appears on average 25 times, 36 times and 39 times respectively (see columns WH in Table 4.2), but near 50% of words in all corpus appear only once (percentage of Hapax). Removing Hapax from the analysis increases the value of the average frequency in all the corpus to 45, 69 and 78, respectively. These datasets are characterised by similar features and are characterised by a significant lexical richness. However, the D2 dataset, despite being half the number of documents of dataset D3, has a larger average document length. The richness of the two lexicons however remains comparable. In the other sections, we present the statistical features of the other datasets used to evaluate ESCAPE.

| **Features** | **WH** | **WoH** | **WH** | **WoH** | **WH** | **WoH** |
|---|---|---|---|---|---|---|
| Dataset ID | D1 | | D2 | | D3 | |
| # categories | 5 | | 10 | | 10 | |
| # documents | 990 | | 2,469 | | 4,939 | |
| Max frequency | 5,394 | | 13,344 | | 19,546 | |
| Min frequency | 1.0 | 2.0 | 1.0 | 2.0 | 1.0 | 2.0 |
| Avg frequency | 25 | 45 | 36 | 69 | 39 | 78 |
| Avg document length | 852 | 836 | 970 | 957 | 705 | 697 |
| # terms | 843,967 | 828,372 | 2,395,721 | 2,363,958 | 3,486,016 | 3,442,508 |
| Dictionary \|V\| | 33,635 | 18,040 | 65,629 | 33,866 | 87,419 | 43,911 |
| TTR | 0.04 | 0.03 | 0.02 | 0.01 | 0.03 | 0.01 |
| Hapax % | 46.3 | 0.0 | 48.2 | 0.0 | 49.1 | 0.0 |
| Guiraud Index | 36.61 | 19.82 | 42.40 | 22.02 | 46.82 | 23.66 |

Table 4.2 Statistical features for the Wikipedia collections.

### 4.1.2 Twitter

Twitter is an American microblogging service on which users post and interact with messages known as *tweets* [167]. Tweets were originally restricted to 140 characters, but on November 7, 2017, this limit was doubled [168]. Twitter can be crawled to extract subsets of tweets related to specific topic. We experimentally validated ESCAPE on a crisis tweet collection [169] containing 60,005 tweets with 16,345 distinct words. Tweets are collected across 6 large events in 2012 and 2013[2]. Thus, the dataset includes 10,000 tweets for each natural disaster and each tweet is labelled with relatedness (i.e., "*on-topic*" or "*off-topic*"). In our analysis, we remove the a-priori knowledge of each label, in order to understand if ESCAPE is able to eliminate the noise present in the collection.

In Table 4.3 the statistical features computed by ESCAPE are reported. With respect to the previous datasets, different values can be seen. Recall that tweets are characterised by shorter lengths of individual documents. In this way the lexical wealth of the collection is less. However, the percentage of hapax is greater, because the words chosen by the individual authors of each twitter are different from person to person. The average document length decreases considerably, as does the average frequency. However, the corpus presents the collection with the largest number of documents analysed.

[2]2012 Sandy Hurricane, 2013 Boston Bombings, 2013 Oklahoma Tornado, 2013 West Texas Explosion, 2013 Alberta Floods and 2013 Queensland Floods

| Features | WH | WoH |
|---|---|---|
| Dataset ID | D4 | |
| # categories | 6 | |
| # documents | 60,005 | |
| Max frequency | 6,936 | 6,936 |
| Min frequency | 1.0 | 2.0 |
| Avg frequency | 19 | 36 |
| Avg document length | 5 | 5 |
| # terms | 312,718 | 304,666 |
| Dictionary \|V\| | 16,345 | 12,136 |
| TTR | 0.05 | 0.03 |
| Hapax % | 49.26 | 0.0 |
| Guiraud Index | 29.23 | 15.02 |

Table 4.3 Statistical features for the Twitter collection.

### 4.1.3 PubMed

PubMed is a free service of the US National Library of Medicine which provides free access to MEDLINE[3]. PubMed is an interface to MEDLINE, the largest biomedical literature database in the world. In scientific literature databases, the identification of relevant predicates between co-occurring concepts is really crucial for using these sources for knowledge extraction [170]. PubMed contains citations and abstracts for more than 25 million articles; PubMed Central also provides full text links [171]. English-language abstracts are as supplied by the publisher or taken directly from the published article. Original policy on inclusion of abstracts set a limit of 250 words for acceptance. We extracted several abstracts from MEDLINE, which are not labelled and for which the real number of categories is not known a-priori. From this large collection, we extract two dataset, which statistics are reported in Table 4.4. We can notice that the two datasets report different data distribution. 1000 papers (including abstracts and sections) have been extracted to generate dataset D5, while D6 represents a large collection characterised by short documents, including about 2500 abstracts. On the other side, we have a small collection in number of documents (i.e. dataset D5), which are characterised by a greater lexical richness as they represent entire papers and not just abstracts of the PubMed collection. The hapax rate is obviously higher in the dataset D5 than D6, however the TTR is

[3]https://www.ncbi.nlm.nih.gov/pubmed/

| **Features** | **WH** | **WoH** | **WH** | **WoH** |
|---|---|---|---|---|
| Dataset ID | D5 | | D6 | |
| # documents | 1,000 | | 2,486 | |
| Max frequency | 775 | | 3,278 | |
| Min frequency | 1.0 | 2.0 | 1.0 | 2.0 |
| Avg frequency | 15 | 18 | 17 | 26 |
| Avg document length | 3600 | 3469 | 103 | 101 |
| # terms | 3,600,153 | 3,469,305 | 257,003 | 251,657 |
| Dictionary \|V\| | 227,210 | 96,362 | 14,818 | 9,472 |
| TTR | 0.06 | 0.05 | 0.06 | 0.04 |
| Hapax % | 57.02 | 0 | 36.07 | 0.0 |
| Guiraud Index | 119.75 | 51.73 | 29.23 | 18.88 |

Table 4.4 Statistical features for the PubMed collections.

comparable in both collections. The number of expected categories is not a-priori known.

### 4.1.4 Reuters

The Reuters dataset, publicly available and known as *Reuters-21578*[4], is a widely used test collection originally created in 1987 by the Carnegie Group, Inc. and Reuters, Ltd for text categorisation research purposes, and therefore made available for research purposes. This dataset is often used for information retrieval, machine learning, and corpus-based researches. The original dataset is made of 21578 articles but in this thesis only a subset of documents has been taken into account. This subset has been created from the *Apte' Split 90 categories*[5], a formatted version of Reuters-21578, that divides the dataset in different categories. The subset used for this study is the whole *Apte' Split 90 categories*, created merging together the test and the training part, for a total of 15.437 documents. The statistical features extracted by ESCAPE are reported in Table 4.5. The features highlight that the Reuters collection is a trade-off between the other corpora. As a matter of fact, it is represented by documents with medium length and a discrete vocabulary. It is not large as the Wikipedia collections, but not too gaunt as the Tweeter collection. However, the lexical richness is rather low.

[4] http://www.daviddlewis.com/resources/testcollections/reuters21578
[5] Dataset on-line available at http://disi.unitn.it/moschitti/corpora.htm

| Features | WH | WoH |
|---|---|---|
| Dataset ID | D7 | |
| # documents | 15,437 | |
| Max frequency | 42,886 | 42,886 |
| Min frequency | 1.0 | 2.0 |
| Avg frequency | 55 | 76 |
| Avg document length | 87 | 85 |
| # terms | 1,337,225 | 1,316,988 |
| Dictionary \|V\| | 24,239 | 17,153 |
| TTR | 0,02 | 0.01 |
| Hapax % | 29.2 | 0.0 |
| Guiraud Index | 20.96 | 14.95 |

Table 4.5 Statistical features for the Reuters collection.

### 4.1.5 Dataset comparison

Through the analysis of the proposed statistical features, we are able to categorise the datasets into few groups according to their statistical indices.

In fact, from the analysis carried out previously, we can observe that the datasets have different characteristics. The Wikipedia documents together with the category PubMed articles are characterised by a greater length and a higher lexical richness than the others, in fact the Guiraud Index is higher for these datasets, reaching the maximum value with the PubMed articles. The dictionary, even after Hapax removal, is extremely high and reflects the complexity of the datasets chosen to test ESCAPE.

On the other side, we have also included two datasets represented by smaller lexical richness, i.e., the Twitter collection and the abstract PubMed collection. The average document length decreases considerably, as does the average frequency. However, the Hapax rate is comparable with the other datasets, and the dictionary after the Hapax removal is smaller with respect to the other datasets. Nevertheless, both the PubMed collections present a further complexity, i.e., the expected number of topics is not known a-priori.

However, among the datasets we have also included the Reuters collection, as it presents differences in data distributions with respect to the other datasets. The Reuters are characterised by a medium length and a lexical index not too high, since the average frequency of the terms is the highest (i.e., the documents are characterised

by a medium length with terms repeated several times). For this reason, the lexical richness is the lowest of all corpora.

## 4.2 Experimental settings

The ESCAPE framework has been developed to be distributed and has been implemented in Python. Since then, all experiments have been performed on the BigData@PoliTO cluster[6] running Apache Spark 2.3.0. The virtual nodes deployed for this research, the driver and the executors, have a 7GB main memory and a quad-core processor each. Below we reported the default configuration for the joint-approach and the default configuration for the probabilistic LDA approach.

**Joint-approach configuration setting.** We recall that for the joint-approach ESCAPE requires two parameters, i.e., the number of dimensions to be considered during the data reduction phase (SVD) and the number of clusters (topics) in which to divide the collection under analysis. During the singular value decomposition reduction phase, the reduction parameter analyses the trend of singular values in terms of their significance. The significance of each dimension is represented by the magnitude of the corresponding singular value. Insignificant dimensions represented by a low magnitude of singular values may represent noise in the data and should be disregarded in the subsequent analysis steps. Thus, we only consider the first $T$ singular values for the analysis. $T$ at most will be equal to the rank of the document-term matrix. This parameter should be set by the analysis, however in our framework we have set this value equal to 20% of the rank. Since the number of documents for all the textual corpora analysed is much smaller than the vocabulary used in each collection, the value $T$ is set by ESCAPE to the 20% of the number of documents. However, the analyst can decide to change the proposed configuration, setting other values for this parameter.

The second parameter that should be set is the number of topics. We propose a new self-tuning algorithm to automatically configure the best configuration. In ESCAPE, the default configuration for the maximum number of cluster is set to the average document length for each corpus. In fact, we hypothesise that every word in the document belongs to at most a different topic. In this way, we set an upper-bound

[6] https://bigdata.polito.it/content/bigdata-cluster

for the value of the number of clusters. However, if the average document length is greater than the number of documents in the corpus under analysis, then the value is set to the average frequency of the term. However, these choices can be changed by each analyst, since the framework being distributed is able to analyse several solutions in parallel.

**Probabilistic model configuration setting.** We recall that for the LDA probabilistic approach, five parameters should be set, which are the maximum number of iterations, the Optimiser, the document concentration ($\alpha$), the topic concentration ($\beta$) and the number of topics (clusters)in which each corpora should be divided. Except for the last parameter, for which we have integrated a self-configuring algorithm, the other four parameters have to be set by the analyst. In ESCAPE the maximum number of iterations within the model has to converge has been set to be equal to 100, the Optimiser (or inference algorithm used to estimate the LDA model) has been set to be Online Variational Bayes. Furthermore, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are set to maximise the log likelihood of the data under analysis. Since we have selected the `Online` optimiser, the $\boldsymbol{\alpha}$ value and the $\boldsymbol{\beta}$ value should be greater than or equal to 0. For this study, the default value for this parameter is $\boldsymbol{\alpha} = 50/K$, as proposed in the literature by different articles [27], [172], [173], and the value set for $\boldsymbol{\beta}$ is $\boldsymbol{\beta} = 0.1$, as proposed in [27].

ESCAPE contains a procedure to relieve the end-user of the burden of selecting proper values for the number of clusters. ESCAPE uses a novel proposed strategy to assess how topics are semantically diverse and choose proper values for the configurations of the probabilistic modelling. As for the joint-approach, in ESCAPE, the default parameter for the maximum number of topic is set to the average document length for each textual collection. Indeed, each word in the document belongs to at most a different topic in our hypothesis. Thus, the upper-bound for the number of topic parameter is set to the average length. However, if the average document length is greater than the number of documents in the corpus under analysis, then the value is set to the average frequency of the term. However, these choices can be changed by each analyst, since the framework being distributed is able to analyse several solutions in parallel.

## 4.3 Joint Approach

In this Section, we report the results obtained using ESCAPE for each dataset for the joint-approach. For each dataset and weighting strategy, ESCAPE provides as output three good configuration of the performed analysis. For each configuration, several quality metrics used to evaluate the goodness of the solutions have been included. To analyse the goodness of each experiment thought the analysis of the main quality metrics integrated, we reported an example of how ESCAPE extracts the three good configurations in Subsection 4.3.1. Then in Subsection 4.3.2, all the experiments done considering the seven corpora are reported. Moreover, in Subsection 4.3.3 an analysis on which each weighting strategy affect the same dataset is described, including also the cardinality of each experiment and the comparison between the different partition found by ESCAPE. Lastly, in Subsection 4.3.4 two datasets have been chosen to show interesting results obtained by the validation and visualisation component. For these datasets, we also reported the impact of each weighting strategy regardless of the chosen approach, to analyse how the different weights are able to emphasize the relevance of words in the corpus.

All the results are related to experiments without excluding the Hapax percentage. As a matter of fact, the removal of Hapax rate in the algebraic approach does not improve performances. This is due to the fact that already in the SVD reduction, the less relevant terms are excluded from the analysis, and therefore the benefit that would result in the analysis is negligible.

### 4.3.1 Top-k solutions

To select the three good configurations to report to the end-user, ESCAPE automatically analyse the quality metrics through a majority model. After the selection of the three values for the SVD reduction, ESCAPE runs several times the K-Means algorithm and compares all the solutions obtained through the analysis of the quality indices.

The ST-DaRe (Self-Tuning Data Reduction) [5] algorithm automatically selects three possible valid values for the LSA parameter in order to identify a good number of dimensions to consider in the subsequent analysis phase without losing significant information. A simple approach, known in the literature, consists in identifying the

maximum decrease point in the curve of singular values. However, this approach could lead to an incorrect choice as it is possible to meet a local minimum.

In ESCAPE, we include an enhanced version of ST-DaRe with only one input parameter to analyse the trend of singular values in terms of significance. The significance of each dimension is represented by the magnitude of the corresponding singular value. Insignificant dimensions represented by a low magnitude of singular values may represent noise in the data and should be disregarded in the subsequent analytics steps. Thus, we only consider the first $T$ singular values for the analysis. Specifically, the mean and the standard deviation values of the magnitude of the first $T$ singular values are computed and then a confidence interval is defined. The selected three-good values of the number of dimensions to consider for the next analytics steps are distributed along the curve: (i) the first is the singular value in correspondence of the mean position, (ii) the second is the singular value in correspondence of the mean plus the standard deviation position, and (iii) the last one is the singular value in correspondence of the mean position of the previous ones. Through this method the problem of the local optimality choice is overcome.

$T$ at most will be equal to the rank of the document-term matrix. However, in our framework we have set this value equal to 20% of the rank. Since the number of documents for all the textual corpora analysed is much smaller than the vocabulary used in each collection, the value $T$ is set by ESCAPE to the 20% of the number of documents.

After that, a majority model is applied to extract the best configuration for each data reduction parameter. We apply a rank function for each quality index used to quantify the goodness of each partition at the variation of the number of clusters. These solutions are compared through the computation of different Silhouette-based quality indices (i.e., Average Silhouette Index, Global Silhouette Index, Weighted Silhouette) defined in Chapter 3. These indices are used to measure the cohesion and separation of each cluster set. Firstly, we define a rank from 2 to the maximum number of clusters, for each index separately. Then, we define a global score function, defined as follow:

$$Score = (1 - rank\_GSI/K_{max}) + (1 - rank\_ASI/K_{max}) + (1 - rank\_WS/K_{max})),$$

where $K_{max}$ is the maximum value of clusters, while *rank_GSI*, *rank_ASI* and $rank\_WS$ are the ranks of the Average Silhouette Index, Global Silhouette Index

| Number of Clusters | GSI | ASI | Weighted - Silhouette | rank_GSI | rank_ASI | rank_WS | Score | Rank-Solution |
|---|---|---|---|---|---|---|---|---|
| 2 | 0.210 | 0.239 | 0.290 | 19 | 18 | 18 | 0.105 | 18 |
| 3 | 0.294 | 0.244 | 0.296 | 16 | 17 | 17 | 0.368 | 17 |
| 4 | 0.255 | 0.237 | 0.290 | 18 | 19 | 19 | 0.053 | 19 |
| 5 | 0.332 | 0.315 | 0.370 | 9 | 4 | 4 | 2.105 | 4 |
| 6 | 0.307 | 0.256 | 0.309 | 14 | 16 | 16 | 0.579 | 16 |
| 7 | 0.383 | 0.354 | 0.405 | 1 | 2 | 2 | 2.737 | 2 |
| 8 | 0.345 | 0.315 | 0.365 | 4 | 5 | 6 | 2.211 | 3 |
| 9 | 0.329 | 0.301 | 0.352 | 11 | 11 | 11 | 1.263 | 11 |
| **10** | **0.383** | **0.357** | **0.409** | **2** | **1** | **1** | **2.789** | **1** |
| 11 | 0.290 | 0.295 | 0.347 | 17 | 12 | 12 | 0.842 | 14 |
| 12 | 0.340 | 0.312 | 0.365 | 5 | 7 | 5 | 2.105 | 4 |
| 13 | 0.336 | 0.306 | 0.358 | 7 | 10 | 10 | 1.579 | 9 |
| 14 | 0.320 | 0.322 | 0.376 | 13 | 3 | 3 | 2.000 | 6 |
| 15 | 0.333 | 0.314 | 0.364 | 8 | 6 | 7 | 1.895 | 7 |
| 16 | 0.336 | 0.311 | 0.363 | 6 | 8 | 9 | 1.789 | 8 |
| 17 | 0.322 | 0.311 | 0.364 | 12 | 9 | 8 | 1.474 | 10 |
| 18 | 0.371 | 0.281 | 0.336 | 3 | 15 | 15 | 1.263 | 11 |
| 19 | 0.330 | 0.284 | 0.337 | 10 | 14 | 14 | 1.000 | 13 |
| 20 | 0.306 | 0.285 | 0.338 | 15 | 13 | 13 | 0.842 | 15 |

Table 4.6 Rank function for dataset D1.

and Weighted Silhouette, respectively. The score lies in the range $[0, (3 - \frac{3}{K_{max}})]$. The worst case is when all the ranks are the smallest for a particular K value, while the highest one is when all the ranks are 1. Lastly, a final rank sort all these scores. ESCAPE selects the best value for each experiment. In Table 4.6, an example is reported. We reported in bold the best configuration found. We also included a plot of the indices' values for each number of clusters in Figure 4.1. In this particular case, $K_{max}$ is equal to 20, so the score lies in the range [0, 2.842].

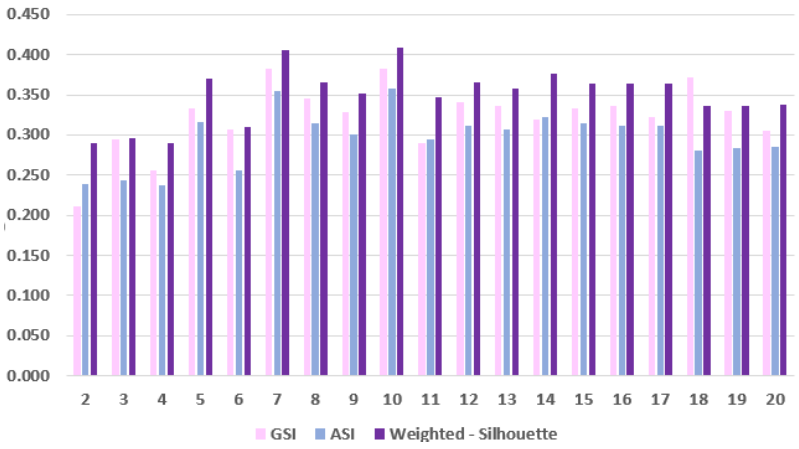



Fig. 4.1 Plot of the silhouette-based indices.

### 4.3.2 Performance

In this Subsection, we report a summary of the experiments conducted on the seven datasets using the joint approach. ESCAPE has been run several times, once for each weighting strategy and dataset. For each dataset, we report the analysis of the main results obtained.

| Dataset | Weight | K-LSA | K-Clustering | GSI | ASI | Weighted - Silhouette | Execution Time |
|---|---|---|---|---|---|---|---|
| D1 | TF-IDF | 26 | 7 | 0.383 | 0.358 | 0.408 | 22m, 20s |
| | | **41** | **10** | **0.419** | **0.339** | **0.391** | |
| | | 67 | 10 | 0.361 | 0.297 | 0.352 | |
| | TF-Entropy | 29 | 11 | 0.334 | 0.350 | 0.401 | 26m, 18s |
| | | **42** | **10** | **0.368** | **0.331** | **0.382** | |
| | | 62 | 8 | 0.364 | 0.274 | 0.326 | |
| | LogTF-IDF | **19** | **5** | **0.437** | **0.431** | **0.480** | 25m, 23s |
| | | 22 | 5 | 0.350 | 0.343 | 0.393 | |
| | | 67 | 4 | 0.225 | 0.201 | 0.251 | |
| | LogTF-Entropy | **10** | **6** | **0.440** | **0.453** | **0.500** | 27m, 12m |
| | | 24 | 5 | 0.323 | 0.318 | 0.367 | |
| | | 67 | 7 | 0.268 | 0.218 | 0.267 | |
| | Bool-IDF | **8** | **5** | **0.445** | **0.444** | **0.494** | 25m, 33s |
| | | 22 | 6 | 0.293 | 0.312 | 0.365 | |
| | | 65 | 6 | 0.226 | 0.233 | 0.286 | |
| | Bool-Entropy | **9** | **5** | **0.447** | **0.444** | **0.495** | 28m, 38s |
| | | 23 | 5 | 0.354 | 0.348 | 0.400 | |
| | | 65 | 4 | 0.280 | 0.234 | 0.285 | |

Table 4.7 Experimental results for dataset D1 for the joint-approach.

| Dataset | Weight | K-LSA | K-Clustering | GSI | ASI | Weighted - Silhouette | Execution Time |
|---|---|---|---|---|---|---|---|
| D2 | TF-IDF | **57** | **13** | **0.280** | **0.236** | **0.288** | 1h, 21m, 16s |
| | | 97 | 18 | 0.181 | 0.168 | 0.222 | |
| | | 165 | 18 | 0.159 | 0.129 | 0.183 | |
| | TF-Entropy | **63** | **13** | **0.271** | **0.209** | **0.265** | 1h, 22m, 39s |
| | | 99 | 18 | 0.219 | 0.172 | 0.226 | |
| | | 161 | 18 | 0.177 | 0.139 | 0.193 | |
| | LogTF-IDF | **25** | **9** | **0.236** | **0.224** | **0.275** | 1h, 27m, 31s |
| | | 56 | 10 | 0.191 | 0.172 | 0.223 | |
| | | 170 | 10 | 0.144 | 0.112 | 0.162 | |
| | LogTF-Entropy | **26** | **7** | **0.270** | **0.233** | **0.281** | 1h, 26m, 26s |
| | | 60 | 6 | 0.230 | 0.173 | 0.224 | |
| | | 169 | 10 | 0.150 | 0.119 | 0.168 | |
| | Bool-IDF | **25** | **9** | **0.221** | **0.213** | **0.263** | 1h, 11m, 18s |
| | | 61 | 7 | 0.208 | 0.166 | 0.216 | |
| | | 180 | 10 | 0.126 | 0.102 | 0.158 | |
| | Bool-Entropy | **26** | **9** | **0.238** | **0.227** | **0.278** | 1h, 40m, 32s |
| | | 62 | 10 | 0.192 | 0.181 | 0.231 | |
| | | 179 | 11 | 0.132 | 0.111 | 0.166 | |

Table 4.8 Experimental results for dataset D2 for the joint-approach.

| Dataset | Weight | K-LSA | K-Clustering | GSI | ASI | Weighted - Silhouette | Execution Time |
|---|---|---|---|---|---|---|---|
| D3 | TF-IDF | **51** | **9** | **0.233** | **0.221** | **0.274** | 1h, 44m, 59s |
| | | 97 | 10 | 0.190 | 0.166 | 0.218 | |
| | | 165 | 10 | 0.158 | 0.125 | 0.179 | |
| | TF-Entropy | **51** | **11** | **0.246** | **0.221** | **0.272** | 1h, 48m, 1s |
| | | 94 | 19 | 0.196 | 0.171 | 0.225 | |
| | | 161 | 17 | 0.172 | 0.134 | 0.189 | |
| | LogTF-IDF | **26** | **9** | **0.220** | **0.205** | **0.255** | 1h, 54m, 15s |
| | | 52 | 8 | 0.183 | 0.158 | 0.208 | |
| | | 150 | 8 | 0.124 | 0.105 | 0.153 | |
| | LogTF-Entropy | **26** | **10** | **0.246** | **0.221** | **0.272** | 1h, 54m, 4s |
| | | 54 | 6 | 0.218 | 0.161 | 0.211 | |
| | | 150 | 7 | 0.140 | 0.110 | 0.157 | |
| | Bool-IDF | **22** | **7** | **0.225** | **0.191** | **0.241** | 2h, 16m, 26s |
| | | 50 | 6 | 0.206 | 0.152 | 0.204 | |
| | | 158 | 5 | 0.155 | 0.091 | 0.141 | |
| | Bool-Entropy | **23** | **6** | **0.257** | **0.196** | **0.247** | 2h, 26m, 58s |
| | | 51 | 10 | 0.269 | 0.177 | 0.227 | |
| | | 158 | 9 | 0.131 | 0.117 | 0.167 | |

Table 4.9 Experimental results for dataset D3 for the joint-approach.

Specifically, from Table 4.7 to Table 4.9 are reported the experimental results obtained for each Wikipedia dataset. Each Table includes the metrics computed for evaluating document partitions for each value selected by our framework. For each dataset and for each weighting strategy, the top-3 solutions (i.e., configurations) are reported to the analyst. However, the best solution among the three identified by ESCAPE is reported in bold.

In more detail, Table 4.7 shows the results obtained for dataset D1. For each weighting strategy, we report the three values used during the data reduction phase, and for each of these values, the best solution is highlighted. The three reduction factors for $K_{LSA}$ are 26, 41 and 67. For each dimensionality reduction parameter, ESCAPE selects the best value for the clustering phase. Given these numbers of dimensions, ESCAPE selects $K-Clustering = 10$ as the optimal partition. We observe that ESCAPE usually selects as optimal partition the experiment exploiting a low-medium number of dimensions (terms). The higher the $K-LSA$, the more variable the data distribution is and the more complex the cluster activity will be. Thus, the Silhouette-based indices tend to slightly decrease when a large number of terms featuring each document (number of dataset columns) is analysed. However, the silhouette-based metrics are quite stable. This means that ESCAPE is able to select only a few main terms to build the model, disregarding the less relevant terms (dimensions).

The TF local weight tends to differentiate the weighted terms, thus identifying a larger number of clusters (associated with different topics in the same category) than the one discovered by the LogTF one. This is also confirmed by the definition itself of the weight. Indeed, the logarithmic function tends to decrease the very high frequency values. In fact, the more the frequency of the term increases, the more the function approaches the asymptote of the logarithm. This means that from a certain frequency, the value of local weight tends to flatten. And the relevance of the most frequent terms is reduced. With respect the global weight instead, the Entropy tends to find in average a large number of clusters.

The TF-IDF and the TF-Entropy find a large number of topics with respect to the other solutions. The other weights instead are able to select the expected value of category. Moreover, the weights TF-IDF and TF-Entropy not only find the original major category but are able to find also the sub-topic related to the major categories. By this way, if the analyst is interested in analysing the dataset at a minor level of detail, he could use these weights, and leave the others for a grain analysis. ESCAPE is able to analyse the same dataset at different granularity levels.

In Tables 4.8 and 4.9 we report the results for the other two datasets of Wikipedia. The comments that we have done before, are also confirmed for these two corpora. Also, in this case, the local weight TF is able to cluster the dataset at a detailed level of description, while the LogTF converges to the expected number of clusters (i.e., ten topics in both cases).

| Dataset | Weight | K-LSA | K-Clustering | GSI | ASI | Weighted - Silhouette | Execution Time |
|---|---|---|---|---|---|---|---|
| D4 | Boolean-IDF | **6** | **6** | **0.465** | **0.422** | **0.737** | 50m, 29s |
| | | 8 | 6 | 0.315 | 0.297 | 0.632 | |
| | | 16 | 10 | 0.237 | 0.181 | 0.413 | |
| | Boolean-Entropy | 7 | 8 | 0.302 | 0.306 | 0.409 | 1h, 10m, 33s |
| | | 9 | 7 | 0.214 | 0.157 | 0.301 | |
| | | **13** | **7** | **0.342** | **0.320** | **0.532** | |

Table 4.10 Experimental results for dataset D4 for the joint-approach.

Different considerations can be discussed for the analysis of the dataset of tweets. Specifically, Table 4.10 reports the results including the main quality metrics. ESCAPE includes three local weights (i.e., Term-Frequency (TF), Logarithmic term frequency (Log) and Boolean term frequency) to highlight the relevance of specific terms in the collection of textual tweets. However, tweets are short messages of at most 140 characters or less. Thus, the number of times that a term occurs in a

document (i.e., term frequency) is often equal to one: a meaningful word is unlikely to be repeated twice in a tweet. In this case, local weighting factor LogTF is equal to TF, and it is trivial demonstrated. Moreover, the only values that TF can assume for each term in a document are 0 (word does not appear in that tweet) or 1 (word does appear in that tweet). Thus, for this dataset, ESCAPE includes as only local weight Boolean measuring either the presence or the absence of each word in each tweet, as reported in Table 4.10.

To compare different configurations, we run ESCAPE once for each combination of weighting function (Boolean as local weight, IDF and Entropy as global weights) together with the LSA reduction method. The top three solutions identified for each weighting strategy and for each number of considered dimensions (LSA reduction factor) are reported in Table 4.10. All selected partitions are good because all the silhouette-based indices assume very high values. As shown in Table 4.10, the identified number of clusters found has a different trend based on the weighting schema used. By increasing the number of dimensions selected through LSA after applying the Boolean-IDF weighting schema, the number of clusters found increases, while with Boolean-Entropy weighting schema, a reverse trend is noted. Moreover, the number of clusters tends to decrease, approaching the expected number of categories (i.e. 6). The Boolean-IDF weighting schema is useful when a more detailed analysis of the categories (disaster type) is of interest because some relevant subtopics within each category are identified, while Boolean-Entropy tends to find the macro-categories at a higher granularity level.

Tables 4.11 and 4.12 show the results produced by ESCAPE for the two PubMed collections. The complexity in validating these results is due to the lack of knowledge of the expected category number. Despite this, however, we can note several trends, already observed previously. The local weights reflect how much face previously; however it can be noted that in the dataset D6, being the analysis of an entire article and not only the abstract, very complex, leads to a choice of a high number of components during the reduction phase. This complexity is further confirmed by the analysis of qualitative indices, lower than the other experiments. While for the D5 dataset, the one relating to the abstracts of the collection, there is almost always a rather low number of categories, except for the weight TF-IDF and TF-Entropy which characterise the group of abstracts in more detail.

| Dataset | Weight | K-LSA | K-Clustering | GSI | ASI | Weighted - Silhouette | Execution Time |
|---|---|---|---|---|---|---|---|
| D5 | TF-IDF | **14** | **5** | **0.352** | **0.284** | **0.333** | 1h, 37m, 19s |
| | | 26 | 7 | 0.258 | 0.229 | 0.281 | |
| | | 54 | 5 | 0.290 | 0.169 | 0.220 | |
| | TF-Entropy | **15** | **10** | **0.377** | **0.280** | **0.332** | 1h, 39m, 34s |
| | | 27 | 16 | 0.247 | 0.204 | 0.254 | |
| | | 52 | 15 | 0.212 | 0.176 | 0.228 | |
| | LogTF-IDF | **15** | **5** | **0.397** | **0.312** | **0.362** | 1h, 43m, 15s |
| | | 28 | 6 | 0.328 | 0.228 | 0.277 | |
| | | 57 | 8 | 0.206 | 0.149 | 0.205 | |
| | LogTF-Entropy | **16** | **5** | **0.384** | **0.287** | **0.336** | 1h, 47m, 34s |
| | | 28 | 5 | 0.303 | 0.177 | 0.228 | |
| | | 57 | 5 | 0.248 | 0.149 | 0.201 | |
| | Bool-IDF | **16** | **4** | **0.315** | **0.347** | **0.395** | 1h, 46m, 42s |
| | | 31 | 5 | 0.285 | 0.265 | 0.319 | |
| | | 60 | 9 | 0.241 | 0.186 | 0.241 | |
| | Bool-Entropy | **16** | **4** | **0.328** | **0.336** | **0.385** | 1h, 48m, 45s |
| | | 31 | 5 | 0.258 | 0.250 | 0.303 | |
| | | 60 | 7 | 0.212 | 0.186 | 0.239 | |

Table 4.11 Experimental results for dataset D5 for the joint-approach.

| Dataset | Weight | K-LSA | K-Clustering | GSI | ASI | Weighted - Silhouette | Execution Time |
|---|---|---|---|---|---|---|---|
| D6 | TF-IDF | **56** | **10** | **0.098** | **0.087** | **0.136** | 35m, 30s |
| | | 102 | 13 | 0.059 | 0.052 | 0.102 | |
| | | 184 | 16 | 0.064 | 0.034 | 0.083 | |
| | TF-Entropy | **59** | **9** | **0.106** | **0.092** | **0.142** | 40m, 54s |
| | | 104 | 10 | 0.080 | 0.050 | 0.099 | |
| | | 184 | 16 | 0.067 | 0.031 | 0.079 | |
| | LogTF-IDF | **33** | **5** | **0.100** | **0.092** | **0.144** | 41m, 14s |
| | | 76 | 5 | 0.058 | 0.051 | 0.103 | |
| | | 182 | 4 | 0.037 | 0.027 | 0.087 | |
| | LogTF-Entropy | **35** | **5** | **0.098** | **0.090** | **0.140** | 49m, 1s |
| | | 80 | 6 | 0.060 | 0.053 | 0.105 | |
| | | 183 | 7 | 0.046 | 0.027 | 0.080 | |
| | Bool-IDF | **24** | **8** | **0.127** | **0.112** | **0.163** | 51m, 49s |
| | | 72 | 6 | 0.056 | 0.054 | 0.107 | |
| | | 190 | 11 | 0.048 | 0.026 | 0.084 | |
| | Bool-Entropy | **26** | **15** | **0.120** | **0.117** | **0.167** | 52m, 19s |
| | | 72 | 15 | 0.060 | 0.057 | 0.109 | |
| | | 189 | 13 | 0.044 | 0.024 | 0.082 | |

Table 4.12 Experimental results for dataset D6 for the joint-approach.

Lastly, we reported the experimental results obtained from the analysis of the Reuters collection, whose main results are reported in Table 4.13. The Reuters collection, together with its earlier variants, has been considered as a standard benchmark for the several text mining activities throughout the last ten years [174]. Indeed, several researches have carved subsets of the original collection, and tested their systems on one of these subsets only. The Reuters collection has proved an extremely popular

| Dataset | Weight | K-LSA | K-Clustering | GSI | ASI | Weighted - Silhouette | Execution Time |
|---|---|---|---|---|---|---|---|
| D7 | TF-IDF | **15** | **10** | **0.246** | **0.257** | **0.159** | 1h, 30m, 58s |
| | | 28 | 13 | 0.220 | 0.206 | 0.132 | |
| | | 59 | 13 | 0.174 | 0.134 | 0.095 | |
| | TF-Entropy | **16** | **14** | **0.254** | **0.256** | **0.157** | 1h, 35m, 23s |
| | | 28 | 9 | 0.259 | 0.199 | 0.132 | |
| | | 60 | 13 | 0.178 | 0.139 | 0.099 | |
| | LogTF-IDF | **16** | **13** | **0.232** | **0.236** | **0.146** | 1h, 34m, 58s |
| | | 27 | 18 | 0.203 | 0.195 | 0.128 | |
| | | 58 | 12 | 0.192 | 0.141 | 0.100 | |
| | LogTF-Entropy | **16** | **10** | **0.229** | **0.238** | **0.150** | 1h, 45m, 56s |
| | | 27 | 14 | 0.207 | 0.190 | 0.126 | |
| | | 59 | 16 | 0.161 | 0.148 | 0.105 | |
| | Boolean-IDF | **13** | **9** | **0.229** | **0.235** | **0.147** | 1h, 43m, 42s |
| | | 27 | 10 | 0.217 | 0.198 | 0.129 | |
| | | 58 | 13 | 0.164 | 0.137 | 0.098 | |
| | Boolean-Entropy | **13** | **10** | **0.220** | **0.223** | **0.143** | 1h, 56m, 41s |
| | | 28 | 11 | 0.212 | 0.198 | 0.131 | |
| | | 59 | 12 | 0.161 | 0.145 | 0.101 | |

Table 4.13 Experimental results for dataset D7 for the joint-approach.

resource and has been used in numerous studies. However, the collection is really complex to analyse, since as described in [175], there are categories which appear in only one document, and many other categories which appear in no documents. In the last past years, researchers are encouraged to include these categories when evaluating the effectiveness of their system, however we can notice in Table 4.13 that the number of categories chosen by ESCAPE are always comparable, independently from the weighting strategy. Moreover, also the quality indices fall in the same ranges, and the number of dimensions considered by ESCAPE are the same. However, despite the complexity of the collection, ESCAPE is able to identify the most relevant partition included in the collection, independently from the weighting strategy chosen.

### 4.3.3 Weight impact

Here, we analyse the impact of the different weighting strategies integrated in ESCAPE. Specifically, to compare the partition found, a matrix $A$ is reported. Each cell $A_{ij}$ in $A$ includes the Adjusted Rand Index (ARI) value obtained by comparing the solutions with weighting strategy $i$ and weighing strategy $j$. The ARI index has as maximum value 1 and its expected value is 0 in case of random clusters. A larger ARI index means a higher agreement between two partitions. The adjusted

Rand index is the corrected-for-chance version of the Rand index. Such a correction for chance establishes a baseline by using the expected similarity of all pair-wise comparisons between clusterings specified by a random model. Since the matrix $A$ is symmetric, we only reported the triangular matrix. We also included a table which summarises the cardinality of each cluster, in terms of number of documents included in each partition, for each dataset under analysis.

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-IDF** | **Boolean-Entropy** |
|---|---|---|---|---|---|
| **TF-IDF** | 0.696 | **0.877** | 0.771 | 0.616 | 0.640 |
| **LogTF-IDF** | | 0.633 | 0.771 | **0.882** | 0.640 |
| **TF-Entropy** | | | 0.706 | 0.559 | 0.578 |
| **LogTF-Entropy** | | | | 0.784 | 0.825 |
| **Boolean-IDF** | | | | | **0.920** |

Table 4.14 Adjusted Rand Index for Dataset D1 for the joint approach.

| | **Cluster ID** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** | **Cluster7** | **Cluster8** | **Cluster9** | **Total** |
| **TF-IDF** | 215 | 176 | 159 | 139 | 99 | 93 | 49 | 25 | 19 | 15 | 989 |
| **TF-Entropy** | 228 | 167 | 166 | 135 | 106 | 75 | 54 | 27 | 16 | 15 | |
| **LogTF-IDF** | 225 | 212 | 191 | 183 | 178 | | | | | | |
| **LogTF-Entropy** | 223 | 191 | 184 | 183 | 105 | 103 | | | | | |
| **Boolean-IDF** | 236 | 223 | 191 | 181 | 158 | | | | | | |
| **Boolean-Entropy** | 230 | 223 | 192 | 177 | 167 | | | | | | |

Table 4.15 Cardinality of each cluster set found for dataset D1 for the joint approach.

In Table 4.14, we report for each couple of weighting strategies, the corresponding ARI found by ESCAPE for dataset D1. We only consider the best solution for each experimental group. There are three experiments that present a very high value for the ARI index: TF-IDF versus TF-Entropy, LogTF-IDF versus Boolean-IDF and Boolean-IDF versus Boolean-Entropy. We recall that ESCAPE includes the ARI metrics since the number of clusters, or the size distribution of those clusters vary drastically. Table 4.15 reports the cardinality of each weighting strategy. Interestingly, for the first pair (i.e., TF-IDF versus TF-Entropy) the number of clusters is similar. In fact, the index in this case confirms the similarity between the partitions, while for the next case (i.e., LogTF-IDF and LogTF-Entropy), despite being the similar number of clusters, the index value decreases. This tells us that the two weights have identified different partitions for the same dataset under analysis. However, despite these lower values, all the partitions obtained, although some differ from others, are quite homogeneous for all the experiments.

In Table 4.16 and Table 4.18, the ARI of all the experiments related to D2 and D3 are reported. Also in this case, we highlight that some results obtained with different

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-IDF** | **Boolean-Entropy** |
|---|---|---|---|---|---|
| **TF-IDF** | 0.544 | 0.618 | 0.632 | 0.471 | 0.466 |
| **LogTF-IDF** | | 0.498 | 0.633 | 0.715 | 0.466 |
| **TF-Entropy** | | | 0.498 | 0.415 | 0.439 |
| **LogTF-Entropy** | | | | 0.664 | 0.661 |
| **Boolean-IDF** | | | | | 0.716 |

Table 4.16 Adjusted Rand Index for Dataset D2 for the joint approach.

| | **Cluster ID** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** |
| TF-IDF | 426 | 330 | 267 | 242 | 231 | 228 | 214 |
| TF-Entropy | 403 | 283 | 277 | 230 | 226 | 219 | 207 |
| LogTF-IDF | 363 | 312 | 295 | 278 | 263 | 247 | 246 |
| LogTF-Entropy | 443 | 348 | 280 | 239 | 235 | 234 | 230 |
| Boolean-IDF | 350 | 332 | 330 | 316 | 261 | 243 | 236 |
| Boolean-Entropy | 363 | 307 | 302 | 296 | 251 | 250 | 249 |
| **TF-IDF** | **Cluster7** | **Cluster8** | **Cluster9** | **Cluster10** | **Cluster11** | **Cluster12** | **Total** |
| TF-Entropy | 199 | 104 | 91 | 91 | 22 | 18 | 2,463 |
| LogTF-IDF | 198 | 168 | 110 | 107 | 22 | 13 | |
| LogTF-Entropy | 230 | 229 | | | | | |
| Boolean-IDF | 227 | 227 | | | | | |
| Boolean-Entropy | 206 | 189 | | | | | |

Table 4.17 Cardinality of each cluster set found for dataset D2 for the joint approach.

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-IDF** | **Boolean-Entropy** |
|---|---|---|---|---|---|
| **TF-IDF** | 0.491 | 0.528 | 0.480 | 0.616 | 0.640 |
| **LogTF-IDF** | | 0.516 | 0.480 | 0.551 | 0.537 |
| **TF-Entropy** | | | 0.528 | 0.350 | 0.318 |
| **LogTF-Entropy** | | | | 0.442 | 0.391 |
| **Boolean-IDF** | | | | | 0.613 |

Table 4.18 Adjusted Rand Index for Dataset D3 for the joint approach.

| | **Cluster ID** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** | **Cluster7** | **Cluster8** | **Cluster9** | **Cluster10** | **Total** |
| **TF-IDF** | 1,313 | 847 | 595 | 534 | 459 | 438 | 298 | 239 | 216 | | | 4,939 |
| **TF-Entropy** | 751 | 589 | 585 | 550 | 439 | 431 | 421 | 419 | 372 | 225 | 157 | |
| **LogTF-IDF** | 669 | 651 | 621 | 611 | 521 | 510 | 468 | 448 | 440 | | | |
| **LogTF-Entropy** | 611 | 596 | 499 | 468 | 465 | 462 | 451 | 438 | 433 | 275 | 241 | |
| **Boolean-IDF** | 995 | 938 | 935 | 622 | 510 | 472 | 467 | | | | | |
| **Boolean-Entropy** | 1,625 | 1,170 | 635 | 536 | 507 | 466 | | | | | | |

Table 4.19 Cardinality of each cluster set found for dataset D3 for the joint approach.

weighting strategies are similar, but less than the previous case. As a matter of fact, for the dataset D2 (See subsection 4.3.4 for further details), ESCAPE emphasises that the solutions obtained with the weighting schema Boolean-IDF and with the Boolean-Entropy are the most similar, followed by the one obtained combining Boolean-IDF and the LogTF-IDF. While for dataset D3, the first similar schemas are the same, followed by the couple TF-IDF and Boolean-IDF and the couple

TF-IDF and Boolean-Entropy. As said before, it could be interesting analysing also the cardinality of the best partitions found by ESCAPE. The two cardinality tables are reported in Table 4.17 and Table 4.19, respectively. D2 and D3 represents larger collection with respect to dataset D1; moreover, the ARI index penalises more partitions with different number of cluster. By this way, the values found in Table 4.16 and Table 4.18 are smaller. As said and described in Section 4.3, the local weight LogTF tends to find a small number of clusters, and this is confirmed by the analysis of the table cardinalities. However, the cluster are quite homogeneous for each experiment, and unbalanced clusters are quite rare.

| | **Boolean-Entropy** |
|---|---|
| **Boolean-IDF** | 0.585 |

Table 4.20 Adjusted Rand Index for Dataset D4 for the joint approach.

| | **Cluster ID** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** | **Total** |
| **Boolean-IDF** | 34,073 | 7,654 | 6,085 | 5,407 | 5,034 | 1,752 | | 60,005 |
| **Boolean-Entropy** | 25,906 | 8,149 | 5,873 | 5,542 | 5,058 | 4,741 | 4,736 | |

Table 4.21 Cardinality of each cluster set found for dataset D4 for the joint approach.

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-IDF** | **Boolean-Entropy** |
|---|---|---|---|---|---|
| **TF-IDF** | 0.625 | 0.673 | 0.613 | 0.550 | 0.570 |
| **LogTF-IDF** | | 0.571 | 0.613 | 0.734 | 0.570 |
| **TF-Entropy** | | | 0.584 | 0.449 | 0.482 |
| **LogTF-Entropy** | | | | 0.685 | 0.732 |
| **Boolean-IDF** | | | | | 0.925 |

Table 4.22 Adjusted Rand Index for Dataset D5 for the joint approach.

| | **Cluster ID** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** | **Cluster7** | **Cluster8** | **Cluster9** | **Total** |
| **TF-IDF** | 443 | 214 | 195 | 124 | 24 | | | | | | 1,000 |
| **TF-Entropy** | 530 | 295 | 288 | 284 | 219 | 213 | 182 | 181 | 154 | 140 | |
| **LogTF-IDF** | 421 | 266 | 146 | 144 | 23 | | | | | | |
| **LogTF-Entropy** | 408 | 274 | 152 | 143 | 23 | | | | | | |
| **Boolean-IDF** | 455 | 274 | 239 | 32 | | | | | | | |
| **Boolean-Entropy** | 441 | 272 | 258 | 29 | | | | | | | |

Table 4.23 Cardinality of each cluster set found for dataset D5 for the joint approach.

For the other datasets, which results are reported from Table 4.20 to Table 4.27, similar considerations can be reported. These tables confirm our hypothesis presented in Section 4.3. Of course when the number of cluster in the same dataset is the same, using different weighting schemas, the ARI value is higher than the case in

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-IDF** | **Boolean-Entropy** |
|---|---|---|---|---|---|
| **TF-IDF** | 0.015 | 0.703 | 0.038 | 0.040 | 0.042 |
| **LogTF-IDF** | | 0.013 | 0.038 | 0.416 | 0.042 |
| **TF-Entropy** | | | 0.041 | 0.043 | 0.041 |
| **LogTF-Entropy** | | | | 0.456 | 0.460 |
| **Boolean-IDF** | | | | | 0.386 |

Table 4.24 Adjusted Rand Index for Dataset D6 for the joint approach.

| | **Cluster ID** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** | **Cluster7** |
| TF-IDF | 407 | 403 | 397 | 307 | 280 | 203 | 179 | 150 |
| TF-Entropy | 503 | 496 | 440 | 309 | 207 | 183 | 167 | 113 |
| LogTF-IDF | 585 | 525 | 519 | 515 | 342 | | | |
| LogTF-Entropy | 491 | 359 | 337 | 324 | 266 | 260 | 158 | 157 |
| Boolean-IDF | 677 | 348 | 328 | 322 | 282 | 243 | 200 | 86 |
| Boolean-Entropy | 254 | 248 | 248 | 224 | 205 | 177 | 158 | 155 |
| **Weight** | **Cluster8** | **Cluster9** | **Cluster10** | **Cluster11** | **Cluster12** | **Cluster13** | **Cluster14** | **Total** |
| TF-IDF | 98 | 62 | | | | | | 2,486 |
| TF-Entropy | 68 | | | | | | | |
| LogTF-IDF | | | | | | | | |
| LogTF-Entropy | 134 | | | | | | | |
| Boolean-IDF | | | | | | | | |
| Boolean-Entropy | 149 | 136 | 131 | 124 | 112 | 87 | 78 | |

Table 4.25 Cardinality of each cluster set found for dataset D6 for the joint approach.

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-IDF** | **Boolean-Entropy** |
|---|---|---|---|---|---|
| **TF-IDF** | 0.541 | 0.660 | 0.627 | 0.433 | 0.449 |
| **LogTF-IDF** | | 0.642 | 0.627 | 0.592 | 0.449 |
| **TF-Entropy** | | | 0.544 | 0.477 | 0.491 |
| **LogTF-Entropy** | | | | 0.569 | 0.572 |
| **Boolean-IDF** | | | | | 0.855 |

Table 4.26 Adjusted Rand Index for Dataset D7 for the joint approach.

| | **Cluster ID** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** | **Cluster7** |
| TF-IDF | 3,963 | 1,819 | 1,690 | 1,517 | 1,247 | 1,233 | 1,115 | 1,005 |
| TF-Entropy | 3,031 | 1,669 | 1,339 | 1,279 | 1,229 | 1,094 | 1,086 | 921 |
| LogTF-IDF | 2,825 | 1,836 | 1,429 | 1,214 | 1,195 | 1,080 | 998 | 980 |
| LogTF-Entropy | 3,593 | 2,255 | 1,686 | 1,594 | 1,455 | 1,295 | 1,249 | 1,055 |
| Boolean-IDF | 3,133 | 3,082 | 1,559 | 1,449 | 1,313 | 1,284 | 1,258 | 904 |
| Boolean-Entropy | 3,162 | 3,155 | 1,740 | 1,535 | 1,463 | 1,353 | 1,314 | 855 |
| **Weight** | **Cluster8** | **Cluster9** | **Cluster10** | **Cluster11** | **Cluster12** | **Cluster13** | **Total** | |
| TF-IDF | 976 | 859 | | | | | 15,424 | |
| TF-Entropy | 910 | 808 | 807 | 745 | 311 | 195 | | |
| LogTF-IDF | 904 | 891 | 878 | 763 | 431 | | | |
| LogTF-Entropy | 859 | 383 | | | | | | |
| Boolean-IDF | 872 | 570 | | | | | | |
| Boolean-Entropy | 847 | | | | | | | |

Table 4.27 Cardinality of each cluster set found for dataset D7 for the joint approach.

which the weighting schemas characterise at different level of detail the dataset. Moreover, the weighting schemas TF-IDF and TF-Entropy have a high value of ARI for all the textual corpora; indeed, these two weighting schemas are able to characterise at a very fine level of detail the corpus. However, also the partitions obtained with the local weight Boolean represent similar partitions. On the other side, the weights that most differentiate the partitions are the TF-Entropy and Boolean-IDF; furthermore, since the cardinality of the partition is completely different, the value of the ARI tends to decrease. However, in all the datasets the ARI index never goes to zero, since at least the main topics includes similar documents. In fact, it is true that the partitions describe the topics at a different level, but when sub-topics are highlighted, there are particular and detailed topics of one of the main topics. In detail, considering both coarse versus fine grained groups, macro-arguments are always identified, independently by the weighting schema used. The difference is in the smaller clusters which characterise the topics. Therefore, ARI almost never reaches levels that are too low, precisely because the clusters are well balanced and the macro categories are identified in all the cases analysed.

### 4.3.4 Visualisation

In this subsection, a subset of interesting results will be reported. In particular, for dataset D2 and dataset D4, we will show different visualisation techniques able to characterise each dataset under analysis, helping the end-user to analyse the different partitions.

#### Dataset D2

Below, the results obtained for dataset D2 (i.e., the 2500 Wikipedia articles collection) are reported. We recall that in this dataset, the expected number of categories is known a-priori and is ten. To analyse the impact of the different weighting strategies, we reported two types of correlation matrices.

First of all, to graphically show the impact of the weighting functions, we analyse the correlation matrix maps in Figure 4.2 for $D2$. Five different coloured correlation ranges have been used for the main bins: 0.87-1.00 black, 0.75-0.87 dark gray, 0.62-0.75 gray, 0.5-0.62 light gray and 0.0-0.5 white. Documents are first sorted by category (which is known for each document), and then the dot products between all

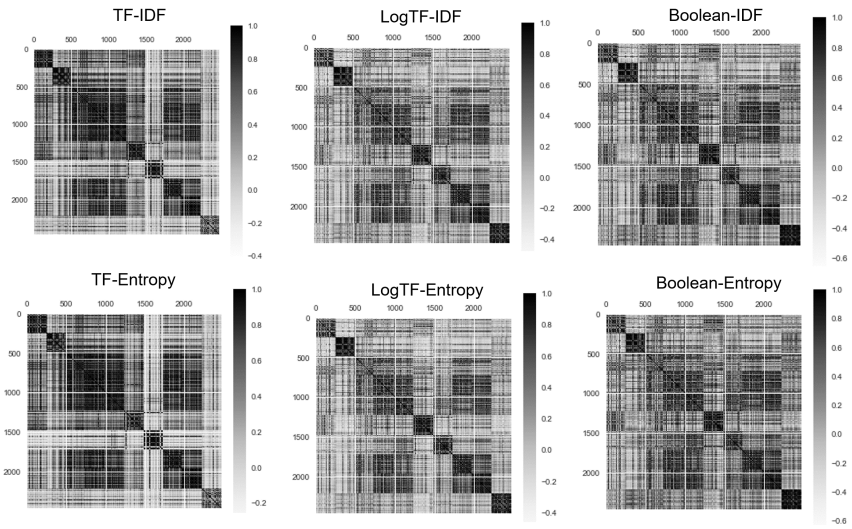



Fig. 4.2 Dataset D2. Correlation matrix maps for analysing the weighting impact.

document pairs are computed. In Figure 4.2, we can for example analysed the impact of the TF-IDF and LogTF-IDF weighting functions respectively on the document collection.

Both functions highlight the 10 macro categories represented as 10 dark rectangles of similar size showing the higher proximity between the documents. Thus, documents belonging to the same macro category tend to be more similar to each other than those belonging to different ones; Log-IDF allows modelling the 10 macro categories better than TF-IDF; whereas TF-IDF highlights possible correlations among different categories.

Instead, Figure 4.3 shows the correlation matrix maps for the best partitions identified by ESCAPE; LogTF-IDF correctly finds the dataset categories whereas TF-IDF also highlights some relevant sub-topics in the same category. As a matter of fact, the local weighting strategy LogTF highlights the main topics, while the TF weight is able to find the sub-topic, which however represent a subset of the main categories. The number of elements per cluster, as described above, is comparable and the results obtained are good in terms of performance.

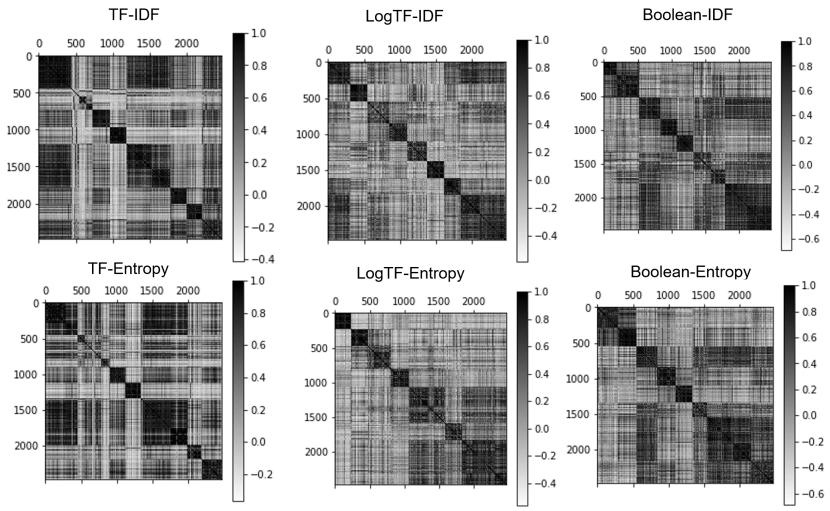



Fig. 4.3 Dataset D2. Correlation matrix maps for analysing the weighting impact for the best configurations.

### Dataset D4

With respect the previous results, here we reported different visualisation results. We recall that D4 is characterise by a collection of Tweets collected across 6 large events in 2012 and 2013. The set of experiments have been designed to show *the effectiveness of* ESCAPE *in discovering a good Tweet partition*.

| Dataset | Weight | K-LSA | K-Clustering | GSI | ASI | Weighted - Silhouette |
|---|---|---|---|---|---|---|
| D4 | Boolean-Entropy | 7 | 8 | 0.302 | 0.306 | 0.409 |
| | | 9 | 7 | 0.214 | 0.157 | 0.301 |
| | | **13** | **7** | **0.342** | **0.320** | **0.532** |

Table 4.28 Subset of experimental results obtained for dataset D4.

In particular, we report in the Table 4.28 the top three solutions identified by ESCAPE for the Boolean-Entropy weighting strategy. We present interesting results obtained for the last experiment, i.e., number of dimensions equals to 13 and number of clusters equals to 7.

For the last experiment, Tables 4.29 and 4.30 show the detailed results, after enriching each tweet with its label (i.e. "on-topic" or "off-topic"). Specifically, we split tweets grouped in each cluster according to label value and category (disaster type). Table 4.29 shows the number of tweets for each cluster and category by only considering the subset of tweets with *off-topic* label, while Table 4.30 is related to *on-topic* label

| | **Off-topic** | | | | | | | **Total number of tweets off-topic** |
|---|---|---|---|---|---|---|---|---|
| | **Cluster ID** | | | | | | | |
| **Category** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| Alberta Floods | 495 | 304 | 53 | 114 | 43 | **3.777** | 49 | 4,835 |
| Boston Bombings | 200 | 114 | 481 | 154 | 53 | **3,309** | 46 | 4,357 |
| Oklahoma Tornado | 285 | 155 | 35 | 604 | 25 | **3,994** | 52 | 5,150 |
| Queensland Floods | 252 | 414 | 34 | 141 | 29 | **3,669** | 68 | 4,607 |
| Sandy Hurricane | 137 | 118 | 34 | 117 | 25 | **3,093** | 343 | 3,867 |
| West Texas Explosion | 190 | 79 | 43 | 159 | 152 | **4,068** | 63 | 4,754 |
| **Tweets for each cluster** | 1,559 | 1,184 | 680 | 1,289 | 327 | 21,910 | 621 | 27,570 |

Table 4.29 Number of tweets for each cluster and category for off-topic label.

| | **On-topic** | | | | | | | **Total number of tweets on-topic** |
|---|---|---|---|---|---|---|---|---|
| | **Cluster ID** | | | | | | | |
| **Category** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| Alberta Floods | **1,715** | **1,837** | 18 | 235 | 23 | **1,342** | 19 | 5.189 |
| Boston Bombings | 185 | 46 | **4,155** | 308 | 125 | 801 | 22 | 5,642 |
| Oklahoma Tornado | 762 | 41 | 6 | **3,686** | 11 | 307 | 11 | 4,824 |
| Queensland Floods | 187 | **4,844** | 3 | 40 | 8 | 258 | 60 | 5,400 |
| Sandy Hurricane | 132 | 146 | 8 | 95 | 21 | 931 | **4,802** | 6,135 |
| West Texas Explosion | 196 | 51 | 188 | 220 | **4,226** | 357 | 7 | 5,245 |
| **Tweets for each cluster** | 3,177 | 6,965 | 4,378 | 4,584 | 4,414 | 3,996 | 4,921 | 32,435 |

Table 4.30 Number of tweets for each cluster and category for on-topic label.

partition. The sum of all tweets in Tables 4.29 and 4.30 forms the complete dataset. $Cluster_6$ mainly includes off-topic tweets (about 80%) and 1,342 tweets related to Alberta Floods.

The other clusters are very similar in numbers of tweets ($Cluster_1$=4,736, $Cluster_2$=8,149, $Cluster_3$=5,058, $Cluster_4$=5,873, $Cluster_5$=4,741 and $Cluster_7$=5,542[7]) and for each of them a main category/topic has been identified (bold numbers in Tables 4.29 and 4.30). For example, $Cluster_2$ is mainly related to *Floods* (both Alberta and Queensland) while $Cluster_5$ describes the *West Texas Explosion* (see Table 4.30). Although the number of clusters is close to the expected value (i.e., six categories), the found partition well separates the main topics.

Since all the best partitions of the tweet collection identified through ESCAPE are anonymous groups of tweets, ESCAPE enhances the explainability of the cluster

[7]These values are obtained by summing the total number of tweets in Tables 4.29 and 4.30 for each cluster

| Cluster 2 | | Cluster 5 | |
|---|---|---|---|
| **Frequent Item** | **Support** | **Frequent Item** | **Support** |
| flood | 0.758 | explos | 0.876 |
| queensland | 0.356 | texa | 0.767 |
| australia | 0.270 | plant | 0.466 |
| water | 0.140 | fertil | 0.382 |
| crisi | 0.103 | west | 0.327 |
| alberta | 0.047 | waco | 0.179 |

Table 4.31 Top 6 items extracted for $\text{Cluster}_2$ and $\text{Cluster}_5$.


Fig. 4.4 WordCloud representations for for $\text{Cluster}_2$ and $\text{Cluster}_5$.

set via the top-k words, based on their frequency (i.e., the percentage of tweets in which each word appears). To this aim, the top-k frequent items (set of words in each cluster characterised by a frequency higher than a given threshold named support) are extracted through the FP-growth algorithm. These words are then represented using the word-cloud technique, to simply show the main topic of each cluster.

Table 4.31 shows the top-6 itemsets (composed of one word) found in $\text{Cluster}_2$ and $\text{Cluster}_5$ by decreasing support (frequency) values. These items well describe the main topic addressed by tweets grouped in the corresponding cluster and they are in line with the results reported in Table 4.30. Moreover, analysing also the word-clouds (reported in Figure 4.4) extracted by the previous clusters, we can better support the confirmation of the goodness of the results obtained. A word cloud [149] is a popular visualisation of words typically associated with textual data. They are most commonly used to highlight salient or relevant terms based on frequency or probability in a collection.

## 4.4 Probabilistic Model

In this Section, we report the results obtained using ESCAPE for each dataset for the probabilistic approach (i.e., Latent Dirichlet Allocation). For each dataset and weighting strategy, ESCAPE selects the top-k good configuration for each number of cluster $K$ obtained as possible good value for the analysis (at most three values), including also several quality metrics used to evaluate the goodness of each solution independently from the weighting strategy adopted. This investigation is reported in Subsection 4.4.1, in which an example of how ESCAPE extracts the good configurations throught the analysis of statistical indices is reported. In Subsection 4.4.2 all the experiments done considering the seven corpora are exposed to analyse the impact of each weighting strategy; moreover in Subsection 4.4.3 is included also the analysis of the cardinality of each experiment and the comparison between the different partition found by ESCAPE. Lastly, in subsection 4.4.4 two datasets have been chosen to show interesting results obtained by the validation and visualisation component.

All the results are related to experiments in which we have excluded the Hapax rate. As a matter of fact, eliminating Hapax allows the probabilistic LDA approach to construct a more precise probabilistic model. Indeed, each term of each textual collection is drawn from its vocabulary, taking into account the terms' probabilities for each given topic of the documents' mixture, excluding words that appear only once within the corpus will allow the reduction of noise in the dictionary.

### 4.4.1 Top-k solutions

In order to find the most suitable number of topics to model a given corpus, in ESCAPE we propose a novel approach, called ToPIC-Similarity [4] (See Chapter 3 for more details). This new strategy assesses how topics are semantically diverse, and then chooses proper configurations for the LDA modelling process. Whit respect to the other state-of-the-art techniques, ToPIC-Similarity is not based on the internal LDA perplexity parameter or on probabilistic quality metrics, but evaluates the topics based on their terms representation. Since the distribution $\Phi$ models the topic-terms distributions in an LDA model, we can extract a description of the topics based on their content. In this way, we can represent each topic content and it is possible to analyse how similar they are each other, choosing a value of K able to

maximise the difference among them. Given a range of cluster values defined by a lower and an upper bound by the analyst (i.e, [$K_{min}$, $K_{max}$]), a new probabilistic LDA model is generated for each $K$. Then, for each of these partitioning processes, ToPIC-Similarity performs three steps, defined as:

1. *topic characterisation*, in which each topic is describe with the most $n$ representative words;
2. *similarity computation*, in which ESCAPE assesses how the topics in the same experiment are similar;
3. *K identification*, in which ESCAPE finds a possible good model configuration to be proposed to the end-user;

For all the $t$ found topics in each $K$ topic model, ESCAPE repeats Steps 1) and 2. All the steps have been described in detail in Chapter 3.

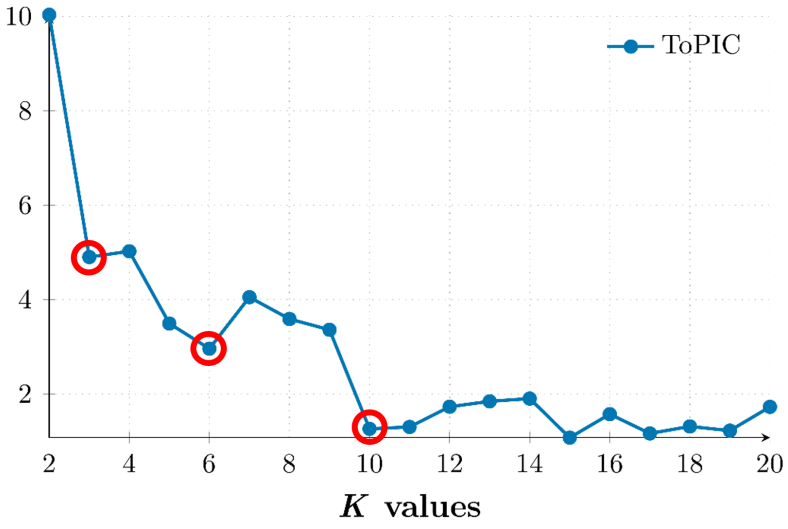



Fig. 4.5 Dataset D1. ToPIC-Similarity curve.

For each experiment, we apply our new methodology to select possible good values for each dataset under analysis. For all the LDA models generated through the various $K$, ESCAPE computes the ToPIC-Similarity measure to define the topic similarity function. To define a trade-off between the optimality and the quality of the results and the execution computation cost, we define this new approach able to find good $K$ values for the analysis. Empirically, we have proved that the obtained

ToPIC-Similarity function is, in most cases, decreasing but not monotonic [4]. The $K$ values are chosen from the occurrence of two conditions:

- points belonging to a local minima of the curve; namely the $K$ values for which $\textit{ToPIC-Similarity}(K_i) < \textit{ToPIC-Similarity}(K_{i+1})$
- points belonging to a decreasing segment of the curve. Thus, ESCAPE considers only those points which have a positive second derivative.

As proposed in [4], we consider the first three points that satisfy both the above conditions as possible good values for the analysis. The topic modelling and the optimal $K$ values search can stop when the first three points are found, or when the upper bound value for the maximum number of clusters is reached by the algorithm (and in this case, a lower number of optimal values will be proposed to the analyst).

An example of the ToPIC-Similarity curve is shown in Figure 4.5, related to dataset D1 for the TF-IDF weighting strategy. The first three points that satisfy both the conditions are 3, 6 and 10.

### 4.4.2 Performance

In this Subsection, we discuss ESCAPE's ability to discover good quality partitions and we present the results and the evaluations obtained through the LDA modelling for the seven corpora that we have used to test ESCAPE. As for the joint-approach, each dataset is evaluated for all the weighting schemas considered in ESCAPE, assessing the goodness of the found results. For each dataset under analysis, we sum up the considerations about the effectiveness of ESCAPE in discovering good partitions, as the different weighting schemas vary.

The main results obtained by ESCAPE for each textual corpus and weighting strategies, are reported from Table 4.32 to Table 4.38. Specifically, Tables from 4.32 to 4.34 are related with the Wikipedia datasets, Table 4.35 with the Tweeter crisis collection. The PubMed results are explored in Tables 4.36 and 4.37. Lastly, the Reuters collection is shown in Table 4.38.

Since the considered weighting schemas highlight the importance of terms within the documents, it could be interesting for the analyst to understand how different weights affect the probabilistic model generated by the LDA. Specifically, for each result

| Dataset | Weight | K | Perplexity | Silhouette | Entropy | Execution Time |
|---|---|---|---|---|---|---|
| D1 | TF-IDF | 3 | 8.8127 | 0.7721 | 0.2561 | 40m, 24s |
| | | 6 | 8.597 | 0.6935 | 0.3634 | |
| | | **10** | **8.4822** | **0.6827** | **0.3956** | |
| | TF-Entropy | **5** | **9.0724** | **0.7623** | **0.2825** | 30m, 32s |
| | | 8 | 9.2482 | 0.6324 | 0.3388 | |
| | | 9 | 9.2679 | 0.6319 | 0.3395 | |
| | LogTF-IDF | **8** | **9.1873** | **0.6754** | **0.3205** | 40m, 17s |
| | | 17 | 9.1262 | 0.637 | 0.3626 | |
| | LogTF-Entropy | 5 | 9.9126 | 0.8915 | 0.1004 | 30m, 54 |
| | | **7** | **9.8841** | **0.846** | **0.1748** | |
| | | 11 | 9.9794 | 0.9515 | 0.1089 | |
| | Boolean-TF | 4 | 6.4926 | 0.6979 | 0.4214 | 44m, 43s |
| | | **5** | **6.464** | **0.6618** | **0.4832** | |
| | | 17 | 6.4208 | 0.3813 | 1.0901 | |

Table 4.32 Experimental results for dataset D1 for the probabilistic approach.

| Dataset | Weight | K | Perplexity | Silhouette | Entropy | Execution Time |
|---|---|---|---|---|---|---|
| D2 | TF-IDF | 3 | 9.2008 | 0.7715 | 0.2460 | 1h, 20m, 23s |
| | | 8 | 8.9628 | 0.5878 | 0.5314 | |
| | | **10** | **8.9436** | **0.5530** | **0.6118** | |
| | TF-Entr | 3 | 9.5568 | 0.8075 | 0.2161 | 1h, 25m, 53s |
| | | **7** | **9.4555** | **0.7008** | **0.3556** | |
| | | 8 | 9.4631 | 0.6985 | 0.3693 | |
| | LogTF-IDF | **11** | **9.4108** | **0.6016** | **0.4895** | 1h, 20m, 46s |
| | | 14 | 9.4529 | 0.5652 | 0.4958 | |
| | LogTF-Entr | **7** | **10.2031** | **0.8751** | **0.1258** | 1h, 10m, 42s |
| | | 9 | 10.2194 | 0.8922 | 0.1219 | |
| | | 11 | 10.2327 | 0.9012 | 0.1253 | |
| | Boolean-TF | 6 | 6.6223 | 0.4398 | 0.7979 | 2h, 20m, 25s |
| | | 13 | 6.5833 | 0.3380 | 1.1922 | |
| | | **18** | **6.5699** | **0.3205** | **1.3262** | |

Table 4.33 Experimental results for dataset D2 for the probabilistic approach.

table, ESCAPE includes a row for each $K$ obtained through the ToPIC-Similarity curve together with the three well-known state-of-the-art quality indices used to explore the goodness of the statistical model generated.

Different trends can be point out and detect from the analysis of these tables. Firstly, we can highlight a reverse linear trend between entropy and silhouette metrics, since better clustering partitions are characterised by a high silhouette value and a small entropy value. Moreover, through the ToPIC-Similarity testing, the TF local weight tends to find in average a smaller number of clusters, independently of the global weight used. On the other hand, the LogTF local weight finds a large number of topics which makes it possible to analyse in a fine detail way the same dataset, since this weight is able to find also some interesting subtopics within the macro-topic. From the exploitation of the global weights, different considerations arise. Indeed, the Global IDF results present a better value for the perplexity index (e.g. at least 0.1 greater) compared to those obtained using global Entropy, although the other quality metrics are not in line.

| **Dataset** | **Weight** | **K** | **Perplexity** | **Silhouette** | **Entropy** | **Execution Time** |
|---|---|---|---|---|---|---|
| D3 | TF-IDF | 3 | 9.001 | 0.243 | 1.181 | 1h, 44m, 4s |
| | | 5 | 8.877 | 0.315 | 1.932 | |
| | | **10** | **8.708** | **0.339** | **2.456** | |
| | TF-Entropy | 5 | 9.120 | 0.202 | 1.783 | 1h, 47m, 48s |
| | | **7** | **9.050** | **0.214** | **1.852** | |
| | | 10 | 9.164 | 0.217 | 2.931 | |
| | LogTF-IDF | 4 | 9.073 | 0.140 | 1.612 | 2h, 19m, 50s |
| | | 14 | 8.921 | 0.192 | 1.753 | |
| | | **16** | **8.917** | **0.198** | **1.819** | |
| | LogTF-Entropy | 3 | 9.444 | 0.060 | 1.564 | 2h, 11m, 43s |
| | | 4 | 9.441 | 0.059 | 1.687 | |
| | | **5** | **9.444** | **0.096** | **2.293** | |
| | Boolean-TF | **11** | **6.309** | **0.220** | **1.902** | 4h, 10m, 45s |

Table 4.34 Experimental results for dataset D3 for the probabilistic approach.

| **Dataset** | **Weight** | **K** | **Perplexity** | **Silhouette** | **Entropy** | **Execution Time** |
|---|---|---|---|---|---|---|
| D4 | Boolean-TF | 4 | 3.490 | 0.513 | 0.594 | 1h, 34m, 31s |
| | | **6** | **2.808** | **0.546** | **0.613** | |
| | | 9 | 3.155 | 0.535 | 0.645 | |

Table 4.35 Experimental results for dataset D4 for the probabilistic approach.

Analysing all the corpora using the Boolean-TF instead, lead to compare very different solutions. This weighting schema is able to find, using our ToPIC-Similarity

curve, three number of topics which represent very different values. Moreover, the first two datasets lead to very high values of silhouette scores, while these values tend to decrease in the other datasets. Indeed, the complexity of the PubMed collections or the Reuters one, imply smaller values of our quality metrics. However, with this methodology, the analyst is able to analyse the same dataset at different granularity levels. For the four datasets for which we know the number of categories (i.e., D1, D2, D3 and D4) the global weight Entropy underestimate the number of topics, finding at least as upper bound the expected number of categories, while the IDF weight tends to overestimate the number of topics. Moreover, the Wikipedia datasets represent the experiments in which the performance found are the highest ones. This behaviour is also confirmed for the other datasets for which we do not know the number of categories.

| **Dataset** | **Weight** | **K** | **Perplexity** | **Silhouette** | **Entropy** | **Execution Time** |
|---|---|---|---|---|---|---|
| D5 | TF-IDF | 3 | 9.715 | 0.285 | 0.208 | 1h, 50m, 27s |
| | | 6 | 9.511 | 0.28 | 0.314 | |
| | | **8** | **9.432** | **0.276** | **0.352** | |
| | TF-Entropy | 4 | 10.115 | 0.28 | 0.225 | 1h, 54m, 25s |
| | | **7** | **10.106** | **0.242** | **0.326** | |
| | | 10 | 10.165 | 0.219 | 0.341 | |
| | LogTF-IDF | 3 | 10.318 | 0.258 | 0.251 | 2h, 14m, 41s |
| | | 4 | 10.229 | 0.283 | 0.293 | |
| | | **6** | **10.164** | **0.301** | **0.301** | |
| | LogTF-Entropy | **3** | **10.823** | **0.114** | **0.101** | 2h, 17m, 25s |
| | | 4 | 10.82 | 0.131 | 0.147 | |
| | | 5 | 10.831 | 0.152 | 0.179 | |
| | Boolean-TF | 5 | 7.782 | 0.283 | 1.108 | 2h, 20m, 13s |
| | | **8** | **7.694** | **0.312** | **1.804** | |
| | | 10 | 7.785 | 0.23 | 1.902 | |

Table 4.36 Experimental results for dataset D5 for the probabilistic approach.

Nevertheless, analysing the goodness of the partitions found only through quantitative metrics is not sufficient, as we limit the analysis to measure the distances (Euclidean and probabilistic) among the groups of documents.

In order to present and make the results obtained more interpretable, different visualisation techniques should be included to represent the main topic in each cluster to effectively validate the probabilistic model. Furthermore, since ToPIC-

| Dataset | Weight | K | Perplexity | Silhouette | Entropy | Execution Time |
|---|---|---|---|---|---|---|
| D6 | TF-IDF | 7 | 7.844 | 0.064 | 1.752 | 1h35m |
| | | 9 | 7.786 | 0.08 | 1.892 | |
| | | **14** | **7.662** | **0.085** | **1.902** | |
| | TF-Entropy | 3 | 8.567 | 0.076 | 1.613 | 1h, 4m, 14s |
| | | **4** | **8.556** | **0.081** | **1.782** | |
| | | 5 | 8.572 | 0.08 | 1.802 | |
| | LogTF-IDF | 7 | 7.925 | 0.092 | 1.613 | 1h, 5m, 17s |
| | | 10 | 7.876 | 0.095 | 1.703 | |
| | | **14** | **7.776** | **0.094** | **1.754** | |
| | LogTF-Entropy | **4** | **8.622** | **0.08** | **1.743** | 1h, 5m, 12s |
| | | 5 | 8.638 | 0.09 | 1.758 | |
| | | 6 | 8.658 | 0.086 | 1.983 | |
| | Boolean-TF | 3 | 5.365 | 0.104 | 1.213 | 1h, 7m, 23s |
| | | 4 | 5.346 | 0.11 | 1.276 | |
| | | **10** | **5.22** | **0.101** | **1.318** | |

Table 4.37 Experimental results for dataset D6 for the probabilistic approach.

| Dataset | Weight | K | Perplexity | Silhouette | Entropy | Execution Time |
|---|---|---|---|---|---|---|
| D7 | TF-IDF | 3 | 7.7154 | 0.7347 | 0.2763 | 55m, 14s |
| | | 4 | 7.6455 | 0.6913 | 0.3485 | |
| | | **9** | **7.4389** | **0.5966** | **0.5586** | |
| | TF-Entropy | 4 | 8.5396 | 0.0564 | 1.3806 | 50m, 35s |
| | | 6 | 8.6242 | -0.247 | 1.7805 | |
| | | **9** | **8.7109** | **-0.0811** | **2.169** | |
| | LogTF-IDF | 5 | 7.7503 | 0.7005 | 0.3565 | 50m, 51s |
| | | 7 | 7.6686 | 0.6676 | 0.444 | |
| | | **13** | **7.5614** | **0.5989** | **0.6396** | |
| | LogTF-Entropy | **5** | **8.788** | **0.0774** | **1.609** | 50m, 19s |
| | | 9 | 9.0011 | -0.0437 | 2.1955 | |
| | | 13 | 9.1759 | -0.069 | 2.56 | |
| | Boolean-TF | 4 | 3.9669 | 0.6373 | 0.4659 | 1h, 20m, 52s |
| | | 7 | 3.8947 | 0.473 | 0.7352 | |
| | | **16** | **3.7309** | **0.3016** | **1.3112** | |

Table 4.38 Experimental results for dataset D7 for the probabilistic approach.

Similarity proposes at most three good values for the topic analysis, the analyst can choose among the various solutions proposed, the one that best reflects the required

granularity of the arguments (i.e., topics). With respect to LSA (the joint-approach), the analysis of only quality metrics is not sufficient to analyse the partitions. A more detailed analysis should be included to help the analyst in interpreting the results. Also, the analysis of how each weighting strategy acts on the LDA model should be analysed to highlight interesting considerations.

### 4.4.3 Weight impact

In the previous section, we have presented the different results that we have obtained analysing only the statistical quality metrics integrated in ESCAPE. Here, we report the different cardinality partitions that we have obtained, and also the Adjusted Rand Index for each couple of weighting schemas. In this way, we are able to understand if a particular weighting strategy is able to improve the results of the LDA model or perform worse despite the good statistical indices. As a result, the analyst is able to learn the different weight impact on the same dataset.

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-TF** |
|---|---|---|---|---|
| **TF-IDF** | 0.299 | 0.466 | 0.118 | 0.671 |
| **LogTF-IDF** | | 0.487 | 0.118 | 0.671 |
| **TF-Entropy** | | | 0.021 | 0.502 |
| **LogTF-Entropy** | | | | 0.108 |

Table 4.39 Adjusted Rand Index for Dataset D1 for the probabilistic approach.

| | Cluster ID | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Weight** | Cluster0 | Cluster1 | Cluster2 | Cluster3 | Cluster4 | Cluster5 | Cluster6 | Cluster7 | Cluster8 | Cluster9 | **Total** |
| **TF-IDF** | 205 | 193 | 187 | 180 | 144 | 21 | 19 | 14 | 13 | 13 | 989 |
| **LogTF-IDF** | 464 | 406 | 91 | 8 | 7 | 5 | 5 | 3 | | | |
| **TF-Entropy** | 428 | 236 | 197 | 113 | 15 | | | | | | |
| **LogTF-Entropy** | 827 | 160 | 1 | 1 | 0 | | | | | | |
| **Bool-TF** | 230 | 215 | 194 | 188 | 162 | | | | | | |

Table 4.40 Cardinality of each cluster set found for dataset D1 for the probabilistic approach.

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-TF** |
|---|---|---|---|---|
| **TF-IDF** | 0.346 | 0.288 | 0.006 | 0.224 |
| **LogTF-IDF** | | 0.356 | 0.006 | 0.224 |
| **TF-Entropy** | | | 0.013 | 0.190 |
| **LogTF-Entropy** | | | | 0.040 |

Table 4.41 Adjusted Rand Index for Dataset D2 for the probabilistic approach.

For each dataset, a similar trend can be reported. Analysing all the ARI comparison for each dataset, an analyst can observe that the LogTF-Entropy weight is the

| | Cluster ID | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** | **Cluster7** | **Cluster8** |
| TF-IDF | 611 | 449 | 340 | 228 | 226 | 193 | 149 | 134 | 95 |
| TF-Entropy | 1,118 | 461 | 343 | 292 | 229 | 12 | 8 | | |
| LogTF-IDF | 798 | 587 | 523 | 350 | 102 | 29 | 22 | 21 | 15 |
| LogTF-Entropy | 2,420 | 31 | 4 | 4 | 3 | 1 | | | |
| Bool-TF | 309 | 266 | 262 | 253 | 248 | 210 | 197 | 192 | 178 |
| **Weight** | **Cluster9** | **Cluster10** | **Cluster11** | **Cluster12** | **Cluster13** | **Cluster14** | **Cluster15** | **Cluster16** | **Total** |
| TF-IDF | 38 | | | | | | | | 2,463 |
| TF-Entropy | 9 | 7 | | | | | | | |
| LogTF-IDF | 137 | 119 | 48 | 24 | 9 | 7 | 4 | 0 | |
| LogTF-Entropy | | | | | | | | | |
| Bool-TF | | | | | | | | | |

Table 4.42 Cardinality of each cluster set found for dataset D2 for the probabilistic approach.

one with lower values when we compare its partition with respect to the other partitions. For example, analysing Table 4.41 and Table 4.48, we can observe that the ARI index for both experiments is the lowest. This means that this weighting strategy is not able to help the LDA topic modelling in find homogeneous partitions. Moreover, analysing the respectively cardinalities, which are descending sorted, we can highlight that the partitions computed using this weighting schema are the most inhomogeneous. There is always a very large cluster, which includes more than 80% of documents. Furthermore, LogTF-Entropy allows to generate empty or singleton clusters with LDA modelling (i.e., clusters with only one document). In particular in Table 4.57 LDA finds two singleton clusters and an empty cluster for dataset D1; while in Table 4.42, Table 4.45 and Table 4.49, which are related to dataset D2, D5 and D6 respectively, at least a singleton cluster has been found. Of course, this kind of partition is not useful for the analyst, as a matter of fact it is like to categorise all the documents in a single macro topic, which represent the union of all the other categories. Also, for the other weighting schemas associated with the global Entropy weight, we can highlight that the number of partitions includes a larger cluster with respect to the others, even if smaller than the one created by the LogTF-Entropy.

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-TF** |
|---|---|---|---|---|
| **TF-IDF** | 0.186 | 0.121 | 0.001 | 0.194 |
| **LogTF-IDF** | | 0.003 | 0.002 | 0.063 |
| **TF-Entropy** | | | 0.013 | 0.055 |
| **LogTF-Entropy** | | | | -0.001 |

Table 4.43 Adjusted Rand Index for Dataset D3 for the probabilistic approach.

A different trend can be presented for dataset D4 and D7 in which a more homogeneous partition is found. See 4.4.4 for more detail related to dataset D7. In these experiments all the partition cardinalities, independently from the weighting schemas

| | Cluster ID | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** | **Cluster7** | **Cluster8** |
| **TF-IDF** | 2,680 | 2,676 | 1,986 | 1,741 | 1,555 | 1,459 | 1,370 | 1,269 | 688 |
| **TF-Entropy** | 3830 | 1068 | 22 | 5 | 5 | 5 | 4 | | |
| **LogTF-IDF** | 2079 | 987 | 791 | 727 | 124 | 56 | 44 | 36 | 34 |
| **LogTF-Entropy** | 4929 | 5 | 3 | 2 | | | | | |
| **Bool-TF** | 755 | 672 | 637 | 589 | 565 | 383 | 379 | 358 | 311 |

| **Weight** | **Cluster9** | **Cluster10** | **Cluster11** | **Cluster12** | **Cluster13** | **Cluster14** | **Cluster15** | **Total** |
|---|---|---|---|---|---|---|---|---|
| **TF-IDF** | 107 | | | | | | | 4,939 |
| **TF-Entropy** | | | | | | | | |
| **LogTF-IDF** | 15 | 13 | 10 | 7 | 7 | 6 | 3 | |
| **LogTF-Entropy** | | | | | | | | |
| **Bool-TF** | 148 | 142 | | | | | | |

Table 4.44 Cardinality of each cluster set found for dataset D3 for the probabilistic approach.

chosen, present cluster that are comparable in number of documents. Moreover, the trend of the ARI of these datasets, is comparable for all the weighting schemas.

| | **Cluster ID** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Weight** | Cluster0 | Cluster1 | Cluster2 | Cluster3 | Cluster4 | Cluster5 | Total |
| **Boolean-TF** | 17,371 | 11,939 | 11,071 | 8,564 | 5,760 | 5,300 | 60,005 |

Table 4.45 Cardinality of each cluster set found for dataset D4 for the probabilistic approach.

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-TF** |
|---|---|---|---|---|
| **TF-IDF** | 0.115 | 0.008 | 0.003 | 0.145 |
| **LogTF-IDF** | | 0.023 | 0.008 | 0.432 |
| **TF-Entropy** | | | 0.279 | 0.002 |
| **LogTF-Entropy** | | | | 0.000 |

Table 4.46 Adjusted Rand Index for Dataset D5 for the probabilistic approach.

| | **Cluster ID** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Weight** | Cluster0 | Cluster1 | Cluster2 | Cluster3 | Cluster4 | Cluster5 | Cluster6 | Cluster7 | Total |
| **TF-IDF** | 355 | 204 | 112 | 103 | 95 | 61 | 36 | 34 | 1000 |
| **TF-Entropy** | 982 | 7 | 4 | 2 | 2 | 2 | 1 | | |
| **LogTF-IDF** | 497 | 397 | 66 | 14 | 14 | 12 | | | |
| **LogTF-Entropy** | 997 | 2 | 1 | | | | | | |
| **Bool-TF** | 449 | 277 | 215 | 26 | 23 | 7 | 3 | | |

Table 4.47 Cardinality of each cluster set found for dataset D5 for the probabilistic approach.

On the other hand, analysing the previous section, we have notice that the IDF global weight is the one which is able to find a number of clusters higher than the expected value. Indeed, from the analysis of both the cardinalities and the corresponding ARI, we can observe that in mean the value of the ARI is the highest for the two weighting

strategies associated with this global weight. Moreover, the partitions found are more homogeneous in terms of number of documents. Instead, for datasets with an average length smaller (e.g. the Tweeter collection D4, or the Reuters collection) the cardinality is almost comparable, regardless of the weight chosen.

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-TF** |
|---|---|---|---|---|
| **TF-IDF** | 0.487 | 0.024 | 0.000 | 0.162 |
| **LogTF-IDF** | | 0.024 | 0.000 | 0.174 |
| **TF-Entropy** | | | 0.016 | 0.016 |
| **LogTF-Entropy** | | | | 0.000 |

Table 4.48 Adjusted Rand Index for Dataset D6 for the probabilistic approach.

| | **Cluster ID** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** | **Cluster7** |
| **TF-IDF** | 436 | 334 | 328 | 314 | 282 | 144 | 119 | 109 |
| **TF-Entropy** | 2,266 | 189 | 17 | 14 | | | | |
| **LogTF-IDF** | 466 | 390 | 390 | 349 | 294 | 260 | 103 | 51 |
| **LogTF-Entropy** | 2,484 | 1 | 1 | | | | | |
| **Bool-TF** | 685 | 358 | 312 | 296 | 257 | 219 | 122 | 117 |

| **Weight** | **Cluster8** | **Cluster9** | **Cluster10** | **Cluster11** | **Cluster12** | **Cluster13** | **Total** |
|---|---|---|---|---|---|---|---|
| **TF-IDF** | 108 | 80 | 68 | 65 | 60 | 39 | 2,486 |
| **TF-Entropy** | | | | | | | |
| **LogTF-IDF** | 45 | 40 | 31 | 28 | 21 | 18 | |
| **LogTF-Entropy** | | | | | | | |
| **Bool-TF** | 63 | 57 | | | | | |

Table 4.49 Cardinality of each cluster set found for dataset D6 for the probabilistic approach.

| | **LogTF-IDF** | **TF-Entropy** | **LogTF-Entropy** | **Boolean-TF** |
|---|---|---|---|---|
| **TF-IDF** | 0.441 | 0.004 | 0.001 | 0.212 |
| **LogTF-IDF** | | 0.006 | 0.001 | 0.213 |
| **TF-Entropy** | | | 0.143 | 0.003 |
| **LogTF-Entropy** | | | | 0.001 |

Table 4.50 Adjusted Rand Index for Dataset D7 for the probabilistic approach.

Specifically, especially when the analyst works with probabilistic approach, analysing only statistical quality metrics is not useful. For this reason, ESCAPE includes all these partition comparisons to help the analyst during the textual data analysis. Moreover, especially for not expert users, we integrate into our engine also several visualisation techniques which are user-friendly and are able to show interesting behaviour in the various datasets.

| | Cluster ID | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Weight** | **Cluster0** | **Cluster1** | **Cluster2** | **Cluster3** | **Cluster4** | **Cluster5** | **Cluster6** | **Cluster7** | **Cluster8** |
| **TF-IDF** | 2,680 | 2,676 | 1,986 | 1,741 | 1,555 | 1,459 | 1,370 | 1,269 | 688 |
| **TF-Entropy** | 2,734 | 2,454 | 1,963 | 1,806 | 1,764 | 1,628 | 1,072 | 1,033 | 970 |
| **LogTF-IDF** | 2,932 | 2,673 | 2,479 | 1,173 | 913 | 912 | 827 | 817 | 665 |
| **LogTF-Entropy** | 3,720 | 3,630 | 2,965 | 2,642 | 2,467 | | | | |
| **Bool-TF** | 3,019 | 2,546 | 2,149 | 1,989 | 1,758 | 883 | 856 | 649 | 415 |

| **Weight** | **Cluster9** | **Cluster10** | **Cluster11** | **Cluster12** | **Cluster13** | **Cluster14** | **Cluster15** | **Total** |
|---|---|---|---|---|---|---|---|---|
| **TF-IDF** | | | | | | | | 15,424 |
| **TF-Entropy** | | | | | | | | |
| **LogTF-IDF** | 632 | 630 | 577 | 194 | | | | |
| **LogTF-Entropy** | | | | | | | | |
| **Bool-TF** | 366 | 249 | 157 | 152 | 125 | 104 | 7 | |

Table 4.51 Cardinality of each cluster set found for dataset D7 for the probabilistic approach.

### 4.4.4 Visualisation

In this subsection, we report a subset of interesting results in a graphical way to ease the knowledge exploitation. In particular, for dataset D5 and dataset D7, we show different visualisation techniques able to characterise each dataset under analysis, helping the end-user to analyse the different partitions. We recall that D5 represents documents of very large size (indeed they are entire paper submitted in MEDLINE), and are characterised by a medium dictionary size; as a matter of fact, the number of terms in the collection is really high, however the number of distinct terms decreases considerably. For this dataset, we report the best solutions found for each weighting strategy, and also the word cloud and the t-SNE representation for analyse into a smaller space the behaviour of our datasets. On the other hand, dataset D7 represents a well-known benchmark for the different text categorisation tasks in the last ten years. The dataset is characterised by documents of small length, featured by a small dictionary, since words appears several times in each document.

#### Dataset D5

In Table 4.52, we report the subset of the results that we have highlighted in bold during the analysis of the statistical quality metrics. For each weighting function, we report the experiment select by ESCAPE as possible good configurations.

We recall that the highest similarity between each couple of weighting strategies is the one composed by the TF-IDF and LogTF-IDF. For the TF-IDF, the number of topics found is 8, while for LogTF-IDF is 6. For each schema, we report the word-clouds of the main topics, the graph representations and the t-SNE visualisations.

| Dataset | Weight | K | Perplexity | Silhouette | Entropy |
|---|---|---|---|---|---|
| D5 | TF-IDF | 8 | 9.432 | 0.276 | 0.352 |
| | TF-Entropy | 7 | 10.106 | 0.242 | 0.326 |
| | LogTF-IDF | 6 | 10.164 | 0.301 | 0.301 |
| | LogTF-Entropy | 3 | 10.823 | 0.114 | 0.101 |
| | Boolean-TF | 8 | 7.694 | 0.312 | 1.804 |

Table 4.52 Experimental results for dataset D5 for the probabilistic approach.

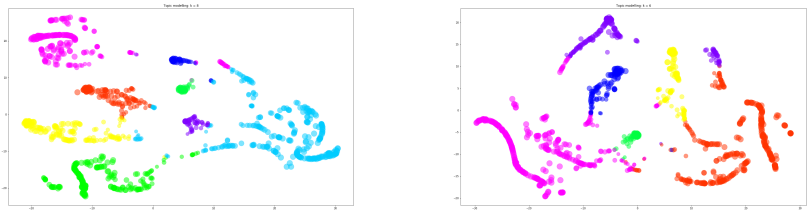
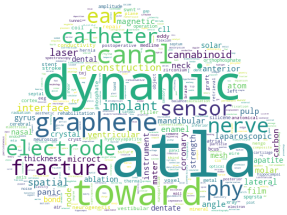


Fig. 4.6 Dataset D5, t-SNE representation. TF-IDF weighting schema (Left) $K$=8 and LogTF-IDF weighting schema (Right) $K$=6.

Firstly, we report in Figure 4.6 the t-SNE representations of the two weighting schemas. The shape is different, but in both case the colours are well balanced.

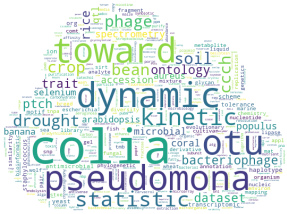
For the weight TF-IDF, we are able to extract the following topics, looking at the word-clouds reported in Figure 4.7. Cluster0 in Figure (4.30a) concerns the analysis of problems related to shock or a feeling of anxiety. The most frequent words (i.e., those reported with a larger size in word-cloud), are related to terms such as *PTSD*, *pediatric*, *geriatric*, *adolescent*. PTSD is a known disorder that arises in some people who have experienced a shocking, frightening or dangerous event. With the typical ups and downs of the emotions and behavior of young children, it is easy to lose delays or problems. In fact, given that nowadays more and more children are exposed to risk factors such as poverty or stress, it increases the likelihood of depression, anxiety, and antisocial behaviour. It has a strong correlation with the Ages and Stages Questionnaire (ASQ), which represents a questionnaire that is compiled by the parents and can be used as a general development screening tool. On the other side, for example Cluster4 in Figure 4.30e is related to cancer analysis. As a matter of fact, we can read in Figure 4.30e as larger words: *BRCA*, which is a gene that normally influences cell growth in the breast but which, when mutated, predisposes to breast cancer. The full name of the gene is breast cancer 1. Another interesting link is represented by *Bcl*, which derives its full name from the B-cell lymphoma 2.


(a) Cluster0

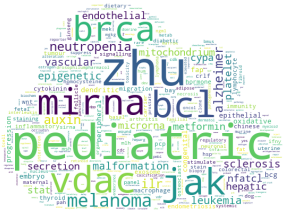



(b) Cluster1

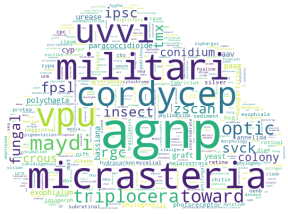

(c) Cluster2



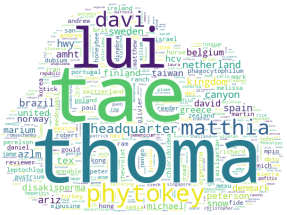

(d) Cluster3



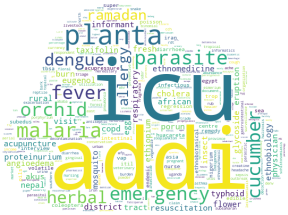

(e) Cluster4




(f) Cluster5




(g) Cluster6




(h) Cluster7

Fig. 4.7 Dataset D5, weighting via TF-IDF. Word-cloud representation.

It represents the second member of a range of proteins which are initially described in chromosomal translocations. Possible damages to the Bcl-2 gene are the basis of numerous cancers, including melanoma, breast, prostate, chronic lymphatic leukemia and lung cancer, but also a possible cause of schizophrenia and autoimmunity. Lastly, an interesting correlation is in the analysis of the VDAC, the Voltage-Dependent Anionic Channel. It is a pore located on the outer membrane of the mitochondrion. For these reasons, VDAC has become a potential therapeutic target to fight cancer but also other diseases in which the mitochondrial metabolism is modified.

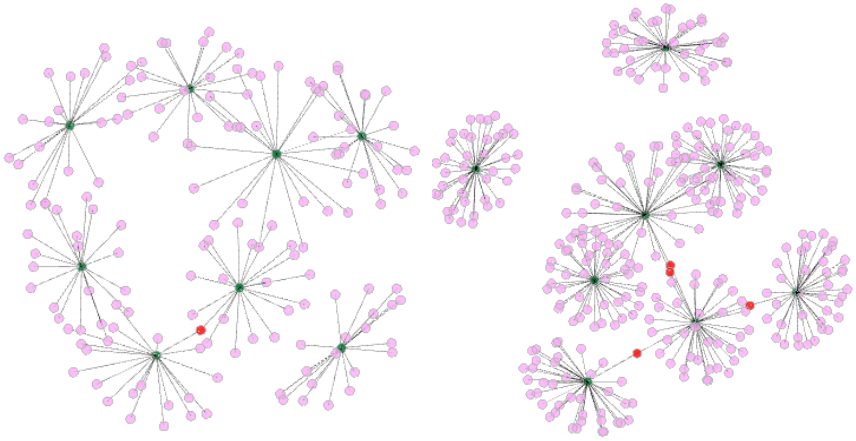
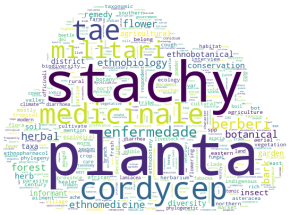


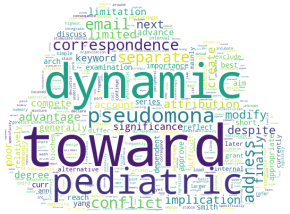
Fig. 4.8 Dataset D5, weighting via TF-IDF. Graph representation. The top-20 most frequent words (Left) and the top-40 most frequent words (Right).

The fact that all the clusters are well separated is also confirmed by the graph representations of dataset D5 using the top-20 and top-40 words (see Figure 4.8). The clusters are well-separated, and the graph is not connected; this means that the words that characterise each topic are different, as already confirmed by the word-clouds. For each topic, we report a dark node, which represents the center of the cluster. For each topic, we extract the top-k most probable words. If a word is in common to multiple topics, we tie the word to each topic, as long as it is one of the most probable top-k words and we color the node in red. Thus, if the clusters are well-separated means that the clusters are characterised by different words and the topics are not overlapped. Conversely, if the word is peculiar to a single topic, we report the corresponding node with a lighter color (pink node). Both graphs in Figure 4.8 are characterised by different words, since only few words appear in more than one cluster.

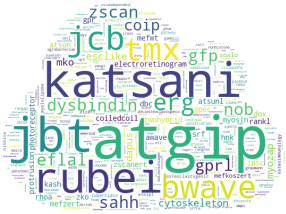



(a) Cluster0

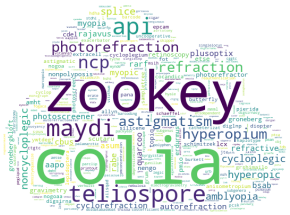

(b) Cluster1



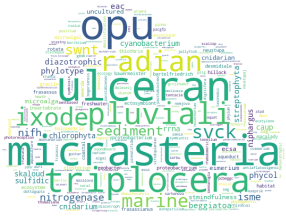

(c) Cluster2




(d) Cluster3




(e) Cluster4




(f) Cluster5

Fig. 4.9 Dataset D5, weighting via LogTF-IDF. Word-clouds representation.

On the other hand, Figure 4.9 reports the word-clouds extracted by the six topics using the LogTF-IDF weighting schema. In details, it is observed that the partitions are different from those previously reported, also we can see that the terms used for the different clusters are different.

However, although the topics found are different, the graphs associated with this local weight, both considering top-20 and top-40 words, remain well separated and not connected. Even with this weight, the words identified for each topic are well separated, as shown in Figure 4.10.

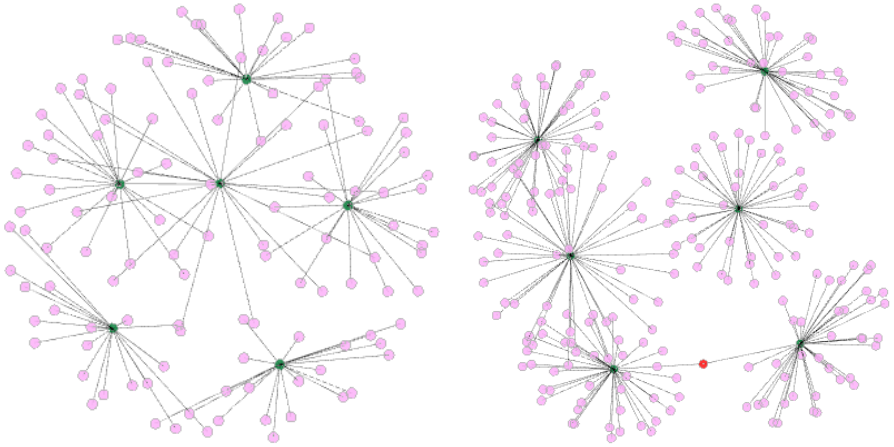


Fig. 4.10 Dataset D5, weighting via LogTF-IDF. Graph representation. The top-20 most frequent words (Left) and the top-40 most frequent words (Right).

### Dataset D7

In Table 4.53, we report the subset of the results obtained by the LDA approach for dataset D7. They represent the selected configuration among the three solutions identified by ESCAPE for each weighting function. Below, the discussion is focused on results obtained for the dataset with TF-IDF, LogTF-IDF and Boolean-TF weighting schemas.

| Dataset | Weight | K | Perplexity | Silhouette | Entropy |
|---|---|---|---|---|---|
| D7 | TF-IDF | 9 | 7.438 | 0.596 | 0.558 |
| | TF-Entropy | 9 | 8.710 | -0.081 | 2.169 |
| | LogTF-IDF | 13 | 7.561 | 0.598 | 0.639 |
| | LogTF-Entropy | 5 | 8.788 | 0.077 | 1.609 |
| | Boolean-TF | 16 | 3.730 | 0.301 | 1.311 |

Table 4.53 Experimental results for dataset D7 for the probabilistic approach.

We recall that ESCAPE provides different graphical visualisations of the obtained dataset partitions to show at a glance the cluster cohesion and separation. The proposed visualisation techniques exploit different kinds of representations to show data and knowledge at different granularity levels. These visualisation techniques allow analysts to easily capture the high-level overview of textual collections through topic detection and clustering algorithms, and drill-down the knowledge to the single document.

Moreover, from the analysis of the ARI index, the two weighting schemas LogTF-IDF and TF-IDF are the most similar partitions with respect to other couples. Firstly, Figure 4.11 shows t-SNE representations, obtained using the TF-IDF and the LogTF-IDF weighting schemas. The t-SNE is a technique able to visualise high-dimensional data over a two dimensional space through a non-linear transformation. This transformation allows similar data points to be represented nearby and, at the same time, different data points to be represented far in the new low-dimensional space. The colouring of the points is based on the assignment to a specific topic, reflecting the results of the topic modelling model. The two t-SNE plots are not the same; however, they represent some similar aspects.

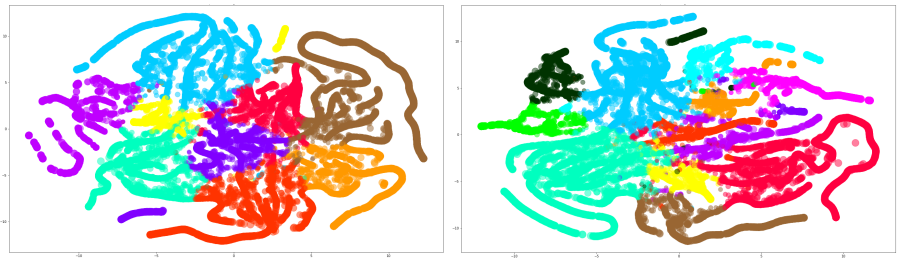


Fig. 4.11 Dataset D7. t-SNE representation. TF-IDF weighting schema (Left) $K$=9 and LogTF-IDF weighting schema (Right) $K$=13.

Indeed, to confirm our hypothesis, we report the graph representations for the two partitions, but also the word-clouds of the top-6 most frequents clusters found by ESCAPE.

While the t-SNE is a useful representation to show the distance between the different documents in a reduced space, the graph is able to represent the topic-term distribution and the compactness and cohesion of the various clusters. In this way, the t-SNE gives us information on how the documents are distributed (ESCAPE selects for each document only the topic with the greatest probability and assigns to each document a specific color, unique for each cluster), while the graph shows us how the terms are distributed in each topic.

Figure 4.12 shows the graphs representations for the weighting strategy TF-IDF (Left) using K=9, and for the weighting strategy LogTF-IDF (Right) using K=13. We have set during the configuration of the graphs, only 20 words. By this way, for each topic the top-20 most probable words have been selected and plotted. Moreover, despite the number is not so high, we can notice that the graph is really hard to analysed. This means that the first 20 most probable words selected by ESCAPE after the

LDA modelling are similar for each topic. This highlights again the complexity of the dataset under analysis. To proof this statement, the graphs obtained including the top-5, the top-10 and the top-40 frequent words for each topic are provided in Figures 4.13, 4.14, 4.15, respectively.

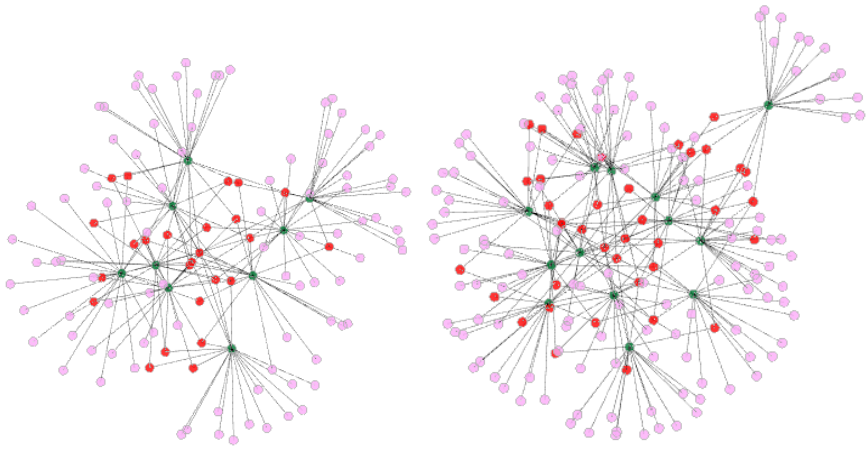


Fig. 4.12 Dataset D7. Graph representation, TF-IDF weighting schema (Left) $K$=9 and LogTF-IDF weighting schema (Right) $K$=13 using the top-20 most frequent words.

We note that the number of words in common for each topic (in the Figure 4.12 are represented by red dots) are quite numerous. In this way, we are sure to state that although you select a few words to characterize each topic, the terms chosen are still present in more than one topic, and this makes clustering difficult.

Even in the simplest case, i.e., considering only 5 words for each topic, we notice intersection between the characterisation of each cluster, as represented in Figure 4.13. In the extreme case, or in this example considering 40 words, we notice that the graph is very connected, and the number of points in common increases considerably, as shown in Figure 4.15.

These analyses are also confirmed by the word-clouds extracted by ESCAPE. Indeed, in Figure 4.16 and in Figure 4.17 we reported the most frequents words for the top-6 frequent clusters in terms of number of documents. The word-clouds present several words which are repeated for these clusters. This means that independently from the weighting strategy, the topic-term distribution is not able to characterise the main topic included in each cluster. The corpus of Reuters is quite complex to divide into well separated topics, as shown by these different visualisations.

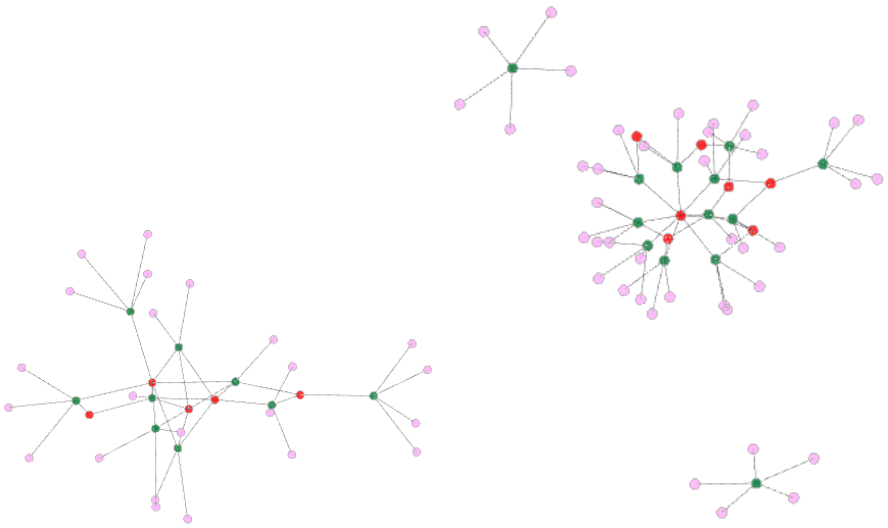


Fig. 4.13 Dataset D7. Graph representation. TF-IDF weighting schema (Left) $K$=9 and LogTF-IDF weighting schema (Right) $K$=13 using the top-5 most frequent words.

Fig. 4.14 Dataset D7. Graph representation. TF-IDF weighting schema (Left) $K$=9 and LogTF-IDF weighting schema (Right) $K$=13 using the top-10 most frequent words.

## 4.5 Comparison

In this Section, the complete set of results obtained by ESCAPE for the representative dataset D1 is presented. We recall here that D1 includes 200 articles for each of the following five categories: *cooking*, *literature*, *mathematics*, *music* and *sport*.

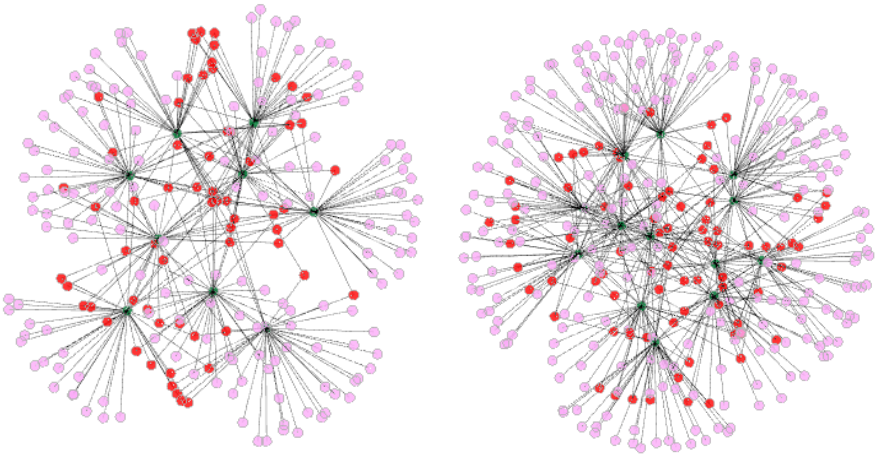


Fig. 4.15 Dataset D7. Graph representation. TF-IDF weighting schema (Left) $K$=9 and LogTF-IDF weighting schema (Right) $K$=13 using the top-40 most frequent words.


Fig. 4.16 Dataset D7, weighting via TF-IDF. Word-cloud representation, $K$=9 for the top-6 most numerous clusters.

Here we reported the two proposed dashboards (i.e. technical and informative dashboards) able to provide to the analysts interesting information at different granularity levels. The technical dashboards are designed to create reports for the domain expert which synthesise data from multiple sources to streamline reporting processes. With their exploitation, the analysts are able to understand how the algorithms work and to analyse the parameter setting of each algorithm in each data analytics phase, including also a comparison with respect to the state-of-the-art approaches. While the informative dashboard includes several graphical representations that are self-explained. These proposed graphical representations are exploited to simplify and


Fig. 4.17 Dataset D7, weighting via LogTF-IDF. Word-cloud representation, $K$=13 for the top-6 most numerous clusters.

synthesise the extracted knowledge patterns in a compact, human-readable, detailed and exhaustive representation. Specifically, we report extracted knowledge analysing the statistical quality metrics used to analyse the different partitions obtained running ESCAPE for each topic modelling approach. However, analysing a corpus considering only quantitative measures is not sufficient. For this purpose, we have proposed several graphs in the form of dashboards useful for exploring the space of the results with innovative and useful visualisation techniques. By this way, the analysts could analyse the different representations integrated in ESCAPE.

### 4.5.1 Technical Dashboard

A technical dashboard includes:

- ESCAPE *performances* for each dataset and methodology. Through the analysis of these results, the analyst is able to compare the impact of each weighting strategy, comparing also the cardinality of each cluster. In particular, we also include the analysis of the weight impact though the computation of *correlation matrices*, useful to analyse the impact of each term in the collection though the different weighting strategies, independently from the methodology.
- *Adjusted Rand Index* to compare the solutions obtained for the same dataset, using different weighting strategies and methodologies.

- *Comparison with the state-of-the-art techniques*, to compare the proposed solution with respect to the well-known state-of-the-art methodologies. For each technique, the main representation have been integrated in ESCAPE.

An analyst can be interest in analysing how some considerations have been done using ESCAPE. Firstly, a deeper comparison between the two methodology could be presented. Indeed, we have seen that the results obtained in the previous sections are described only using quantitative metrics. However, other representations should be presented to highlight the proposed approach. Moreover, a comparison with the well-known state-of-the-art techniques is also included since it could be interesting in a technical dashboard tailored to expert analyst.

### ESCAPE performance

| Dataset | Weight | K-LSA | K-Clustering | GSI | ASI | Weighted - Silhouette |
|---|---|---|---|---|---|---|
| D1 | TF-IDF | 41 | 10 | 0.419 | 0.339 | 0.391 |
| | LogTF-IDF | 19 | 5 | 0.437 | 0.431 | 0.480 |
| | TF-Entropy | 42 | 10 | 0.368 | 0.331 | 0.382 |
| | LogTF-Entropy | 10 | 6 | 0.440 | 0.453 | 0.500 |
| | Bool-IDF | 8 | 5 | 0.445 | 0.444 | 0.494 |
| | Bool-Entropy | 9 | 5 | 0.447 | 0.444 | 0.495 |

Table 4.54 The best ESCAPE results. Dataset D1. Joint-approach.

| Dataset | Weight | K | Perpl | Silh | Entr |
|---|---|---|---|---|---|
| D1 | TF-IDF | 10 | 8.4822 | 0.6827 | 0.3956 |
| | LogTF-IDF | 8 | 9.1873 | 0.6754 | 0.3205 |
| | TF-Entropy | 5 | 9.0724 | 0.7623 | 0.2825 |
| | LogTF-Entropy | 7 | 9.8841 | 0.8460 | 0.1748 |
| | Bool-TF | 5 | 6.4640 | 0.6618 | 0.4832 |

Table 4.55 The best ESCAPE results. Dataset D1. Probabilistic approach.

In Tables 4.54 and 4.55 we report the results obtained for dataset D1. Specifically, Table 4.54 reports the results obtained for the join-approach, while Table 4.55 reports the results obtained for the probabilistic approach.

In Tables 4.56 and 4.57 are reported the cardinalities of the different cluster-sets found by ESCAPE for dataset D1. An analyst can be interested in analysis the difference between the two type of partitions obtained using the two strategies. To this aim,

| | Cluster ID | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Weight** | Cluster0 | Cluster1 | Cluster2 | Cluster3 | Cluster4 | Cluster5 | Cluster6 | Cluster7 | Cluster8 | Cluster9 | **Total** |
| **TF-IDF** | 215 | 176 | 159 | 139 | 99 | 93 | 49 | 25 | 19 | 15 | 989 |
| **TF-Entropy** | 228 | 167 | 166 | 135 | 106 | 75 | 54 | 27 | 16 | 15 | |
| **LogTF-IDF** | 225 | 212 | 191 | 183 | 178 | | | | | | |
| **LogTF-Entropy** | 223 | 191 | 184 | 183 | 105 | 103 | | | | | |
| **Boolean-IDF** | 236 | 223 | 191 | 181 | 158 | | | | | | |
| **Boolean-Entropy** | 230 | 223 | 192 | 177 | 167 | | | | | | |

Table 4.56 Cardinality of each cluster found for dataset D1 for the joint approach.

| | Cluster ID | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Weight** | Cluster0 | Cluster1 | Cluster2 | Cluster3 | Cluster4 | Cluster5 | Cluster6 | Cluster7 | Cluster8 | Cluster9 | **Total** |
| **TF-IDF** | 205 | 193 | 187 | 180 | 144 | 21 | 19 | 14 | 13 | 13 | 989 |
| **LogTF-IDF** | 464 | 406 | 91 | 8 | 7 | 5 | 5 | 3 | | | |
| **TF-Entropy** | 428 | 236 | 197 | 113 | 15 | | | | | | |
| **LogTF-Entropy** | 827 | 160 | 1 | 1 | 0 | | | | | | |
| **Bool-TF** | 230 | 215 | 194 | 188 | 162 | | | | | | |

Table 4.57 Cardinality of each cluster found for dataset D1 for the probabilistic approach.

ESCAPE compares the best solutions found by the two different methodologies and compute the ARI index for the obtained partitions, which give us a quick comparison of the similarity between the partitions.

| | **Weighting schema** | | | | |
|---|---|---|---|---|---|
| **Dataset** | TF-IDF | LogTF-IDF | TF-Entropy | LogTF-Entropy | Boolean |
| D1 | 0.554 | 0.321 | 0.320 | 0.100 | 0.790 |

Table 4.58 Adjusted Rand Index for Dataset D1.

The ARI between the best partitions of the two methodologies is reported in Table 4.58. We can observe that the results are quite different and analysing only the previous table is not sufficient to try conclusions on the two methodologies. Since the Boolean-IDF and Boolean-Entropy are very similar in terms of partitions for the joint-approach, we only consider the weight Boolean-Entropy for the comparison with respect to the Boolean-TF weight.

We recall also that the ARI index penalises more than the Rand Index the partitions with different number of clusters; however, we can notice that specially for the weighting LogTF-Entropy, the comparison value is really poor. A deeper analysis should be presented. To this aim, we compare the weighting schemas TF-IDF and LogTF-Entropy for the two different methodologies, to analyse if there is some behaviour in these approaches.

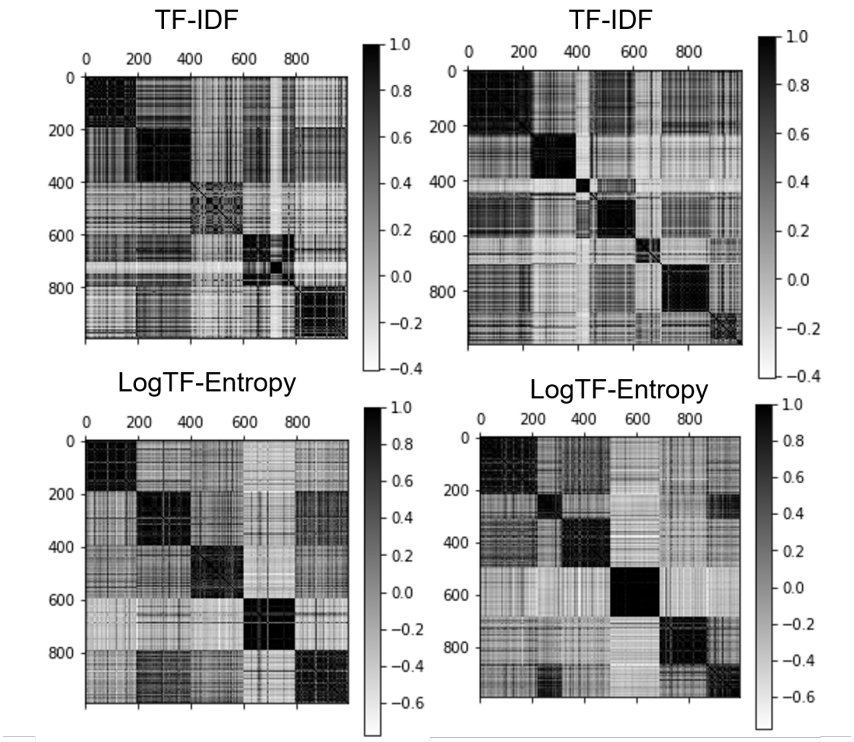



Fig. 4.18 Correlation matrix maps for dataset D1 for analysing: the weighting impact (Left) and the best partitions (Right).

To graphically show the impact of both weighting functions for the joint-approach, ESCAPE analyses the correlation matrix maps reported in Figure 4.18 for $D_1$. Five different coloured correlation ranges have been used: 0.87-1.00 black, 0.75-0.87 dark gray, 0.62-0.75 gray, 0.5-0.62 light gray and 0.0-0.5 white. Documents are first sorted by category, and then the dot products between all document pairs are computed. Figure 4.18(Left) shows the impact of the TF-IDF and LogTF-Entropy weighting functions respectively on the document collection.

Both functions highlight the 5 macro categories represented as 5 dark squares of similar size showing the higher proximity between the documents. Thus, documents belonging to the same macro category tend to be more similar to each other than those belonging to different ones; Log-IDF (Figure 4.18) (Left on the bottom) allows modelling the 5 macro categories better than TF-IDF (Figure 4.18) (Left on the top) and also characterises some topics; whereas TF-IDF highlights possible correlations among different categories.

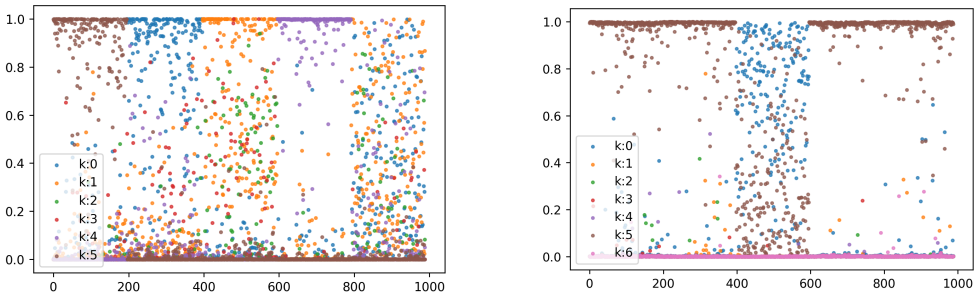



Fig. 4.19 Document probability distributions in each topic for weighting TF-IDF (Top) and LogTF-Entropy.

Figure 4.18 (on the Right) shows the correlation matrix maps for the best partitions identified by ESCAPE; Log-IDF 4.18 (Right on the bottom) correctly finds the dataset categories whereas TF-IDF 4.18 (Right on the top) also highlights some relevant subtopics in the same category.

Since weights highlight the importance of words within documents, analysing how different weighting schemas affect the model is important. For the representative dataset D1, ESCAPE computes the histogram of the TF-IDF and LogTF-Entropy weights. The values of LogTF-Entropy present almost a uniform distribution in the range [0,1] (Kurtosis index > 0 and standard deviation 0.5) and the distribution has maximum value 8. Instead, with the IDF an asymmetrical bell distribution is obtained with average values between [2,5] (Kurtosis index > 0 and standard deviation 12.7) and maximum value 1161. The IDF weight schema better differentiates the weights within the corpus, thus producing a probabilistic model with better performances. The Entropy global weight performs badly in providing relevance for the words in all the datasets. As shown in Figure 4.19, even though the quantitative evaluation metrics cannot spot the bad results of the clustering produced with these weighting schemas, it is still possible to assess the quality of the generated model. Indeed, Figure 4.19 presents interesting considerations analysing dataset D1 for TF-IDF (on the Left) and LogTF-Entropy (on the Right) schemas. Specifically, Figure 4.19 shows for the LDA models the probability distribution of each document of the corpus D1 to belong to the K topics selected by the algorithm (i.e., K=6 for TF-IDF (we used the second-best solution due to the limited number of clusters) and K=7 for LogTF-Entropy). In detail, the documents present a more homogeneous distribution using the IDF weight, with topics balanced by the number of documents. Instead,

with the Entropy weight, there is one cluster in which 90% of the documents have a probability greater than 0.90 of membership. It turns out that 90% of the documents belong to a single cluster (topic) and the result is due to the fact that the weight entropy fails to isolate the most significant terms within the collection of documents.

We can conclude that some weighting strategies are useful for a particular analysis with respect to the others. As a matter of fact, from the analysis of the histograms, and also from the results analysed in the previously, we can assess that the IDF weight schema better differentiates the weights within the corpus, thus producing a probabilistic model with better performances. As for the running example, also for all the other datasets has been observed that the Entropy global weight badly performs in bringing relevance to the words.

This can be explained by the visualisation charts, even if the quantitative evaluation metrics (perplexity, Silhouette and entropy) cannot spot the bad results of the clustering produced with these weighting schemas. Indeed, the probabilistic quantitative metrics evaluate the confidence the model has in assigning the documents the topic labels. In the results obtained with the Entropy global weight, even if erroneously, LDA assigns with a high confidence the document to a topic, thus leading the quality metrics to not spot badly performance of the model in dividing the corpus in different cohesive clusters.

When unbalanced clusters are generated, the use of only goodness metrics is not able to guarantee good performance. Indeed, high values of Silhouette or low values of entropy do not involve a good clustering but represent a simplification of the problem. It is like classifying 90% of the documents in a single topic, thus generating many false negatives. Having the class label available, indices such as recall, or precision could help identify these incorrect assignments. However, if the label were not available, the use of quantitative indicators would not be effective. Methods that take into account the semantics must be presented.

On the other hand, the joint approach leads to better results from the point of view of the partitions. In fact, the weights in this case analyse the same dataset at different levels of detail, without creating unbalanced clusters. In fact, the K-Means algorithm is benefiting from the previous LSA reduction, in this way its performances are far superior.

### State-of-the-art comparison

In the technical dashboard, ESCAPE includes also a comparison with the state-of-the-art techniques, described in Chapter 2.

#### Joint-Approach

To evaluate the effectiveness of ESCAPE in correctly identifying good values for the desired number of clusters, we compared our results with a state-of-the-art method known as the *Elbow graph* or *Knee* approach [105], denoted as $k_{SSE}$ below. This method analyses the trend of SSE (Sum of Squared Errors) against $k_{cls}$. The optimal $k_{cls}$ value must be selected at the coordinates where the gain from adding a centroid is negligible, i.e. the SSE reduction is not interesting any more. Here we discuss $D_1$ as the representative dataset, and the results on all the other datasets follow a similar trend.

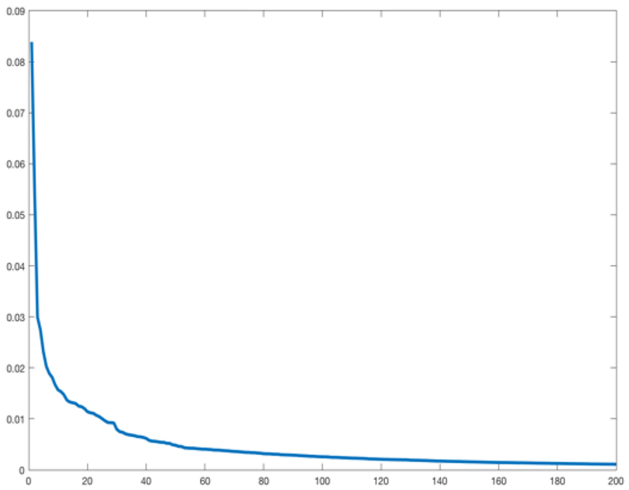



Fig. 4.20 Top singular values for Dataset D1 weighted via LogTF-Entropy.

For a fair comparison both methods, ESCAPE and the analysis through the *Elbow graph*, receive as input the reduced matrix $X_{K-LSA}$. The reduced matrix $X_{K-LSA}$ is obtained analysing the trend of the singular values obtained by the decomposition of the original document-term matrix. In our proposed methodology, ESCAPE selects the possible good values at the points: 10, 24 and 67. As shown in Figure 4.20 these three points are able to characterise the singular value plot, analysing different values which subsequently include a large number of dimensions in the reduction phase.

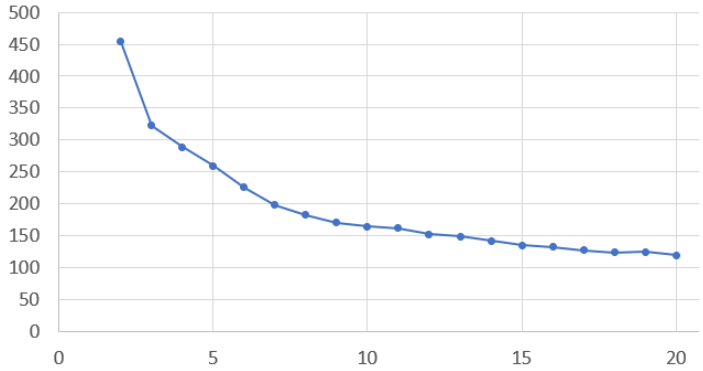



Fig. 4.21 SSE trend for Dataset $D1$ weighted via LogTF-Entropy for the joint approach.

However, the $k_{SSE}$ method usually sets a lower value for the desired number of clusters than the proposed approach. For example, for $D_1$ the $k_{SSE}$ method selects 5 clusters against 10 set by ESCAPE by exploiting TF-IDF, and 3 clusters against 6 with LogTF-Entropy. In Figure 4.21 are reported the analysis of the SSE for the weighting schema LogTF-Entropy.

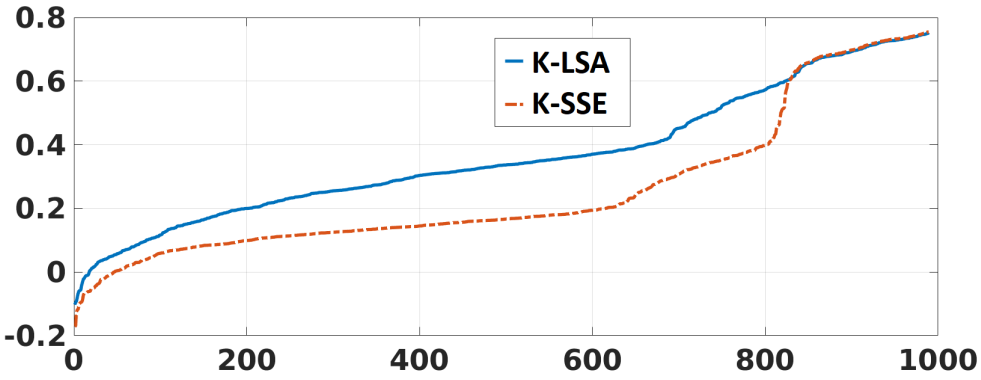



Fig. 4.22 Silhouette index for Dataset $D1$ weighted via LogTF-Entropy for the joint approach.

The two best partitions, discovered via ESCAPE and $k_{SSE}$ by using LogTF-Entropy, are compared by evaluating the Silhouette index for each clustered document (see Figure 4.22). More than 83% of documents have a better placement in the partition discovered by ESCAPE with respect to the one selected through the analysis of the SSE curve. Thus, ESCAPE is able to discover a cluster set better than the Knee approach.

### Probabilistic Approach

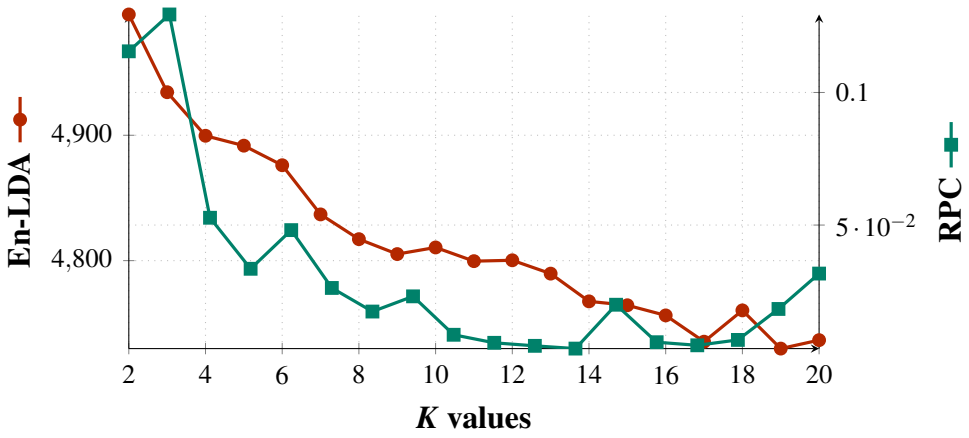
In this Section, a comparison of the results obtained with ESCAPE and state-of-the-art techniques (i.e., *RPC* and *En-LDA*) is presented. RPC [117] is a heuristic algorithm evaluating the average perplexity variation of the LDA models to choose the number of topics. EnLDA [118] is instead an Entropy-based approach which selects the $K$ value in order to minimise the overall amount of entropy of the topic modelling. These techniques will be discussed in more detail below. We integrated two types of comparison: (i) quantitative comparison and (ii) qualitative comparison. Using the quantitative comparison, we include the main statistical features used to analyse the goodness of the partitions, while for the qualitative comparison, the main visualisation techniques described in Chapter 3 are used to analyse the same results in a different and human-readable way.

### Quantitative comparison

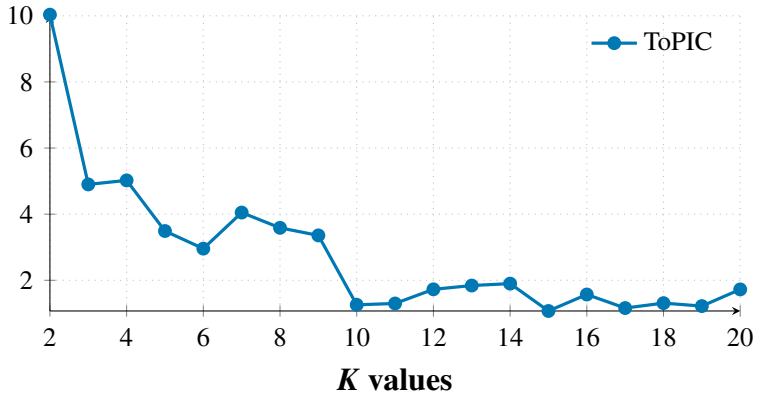
The results obtained by ESCAPE with the TF-IDF weighting schema are reported in Figure 4.23b. From the chart, the selected $K$ values are 3, 6 and 10. While in Figure 4.23a are reported the values obtained by the two state-of-the-art methods.

| | Weights | Method | K | Perpl | Silh | Entr |
|---|---|---|---|---|---|---|
| **D1** | **TF-IDF** | RPC | 3 | 8.812 | 0.772 | 0.256 |
| | | En-LDA | 19 | 8.427 | 0.621 | 0.534 |
| | | ESCAPE | 10 | 8.482 | 0.682 | 0.395 |
| | **TF-Entr** | RPC | 5 | 9.072 | 0.762 | 0.282 |
| | | En-LDA | 5 | 9.072 | 0.762 | 0.282 |
| | | ESCAPE | 5 | 9.072 | 0.762 | 0.282 |
| | **LogTF-IDF** | RPC | 7 | 9.183 | 0.693 | 0.319 |
| | | En-LDA | 16 | 9.189 | 0.553 | 0.443 |
| | | ESCAPE | 8 | 9.187 | 0.675 | 0.320 |
| | **LogTF-Entr** | RPC | 3 | 9.777 | 0.852 | 0.144 |
| | | En-LDA | 3 | 9.777 | 0.852 | 0.144 |
| | | ESCAPE | 7 | 9.884 | 0.846 | 0.174 |
| | **Boolean-TF** | RPC | 4 | 6.492 | 0.697 | 0.421 |
| | | En-LDA | 20 | 6.412 | 0.661 | 1.255 |
| | | ESCAPE | 5 | 6.464 | 0.661 | 0.483 |

Table 4.59 Performance of State-of-the-art methods vs ESCAPE.




(a) Dataset D1, weighted via TF-IDF, En-LDA and RPC results.




(b) Dataset D1, weighted via TF-IDF, ESCAPE results.

Fig. 4.23 En-LDA, RPC and ESCAPE results diagrams for dataset D1, weighted via TF-IDF.

Table 4.59 shows the results obtained with the state-of-the-art techniques with respect to ESCAPE and their evaluation using the same metrics presented previously. It is observable that the first number of topics found by ESCAPE is comparable with the value found by the RPC method (practically equal in D1 using TF-IDF), which tends to find a very small number of topics. Instead, ESCAPE is comparable with En-LDA (which tends to create many clusters, sometimes even taking the upper bound of the possible $K$ values as the optimal solution) using the last value of K and as weighting function the Boolean-TF$_{glob}$. This weight finds a greater number of

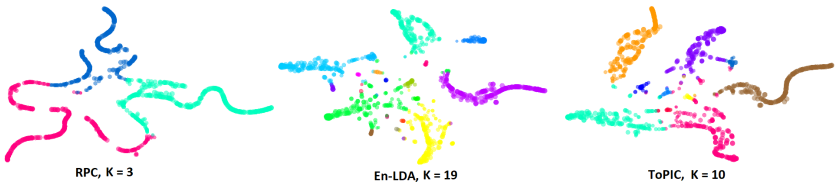



Fig. 4.24 Dataset D1. Comparison of t-SNE representations.

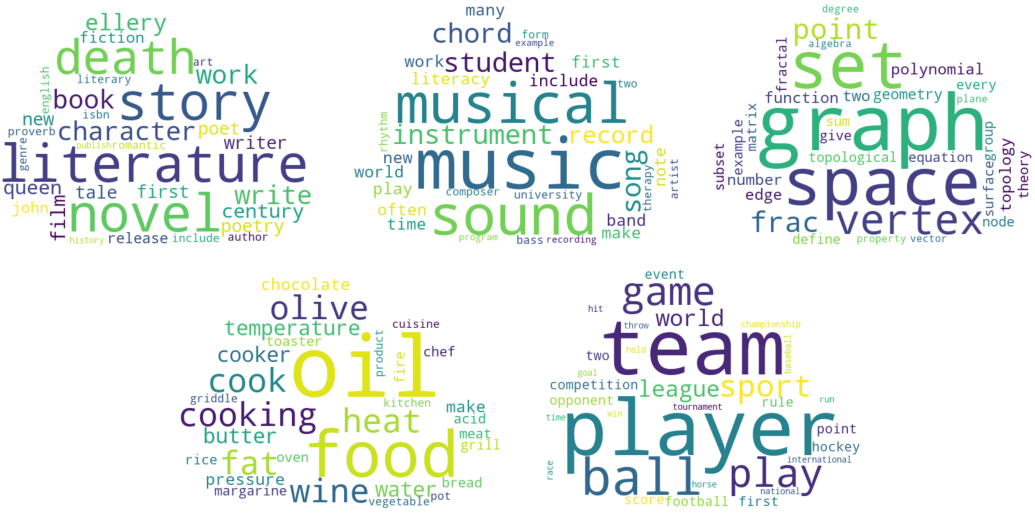
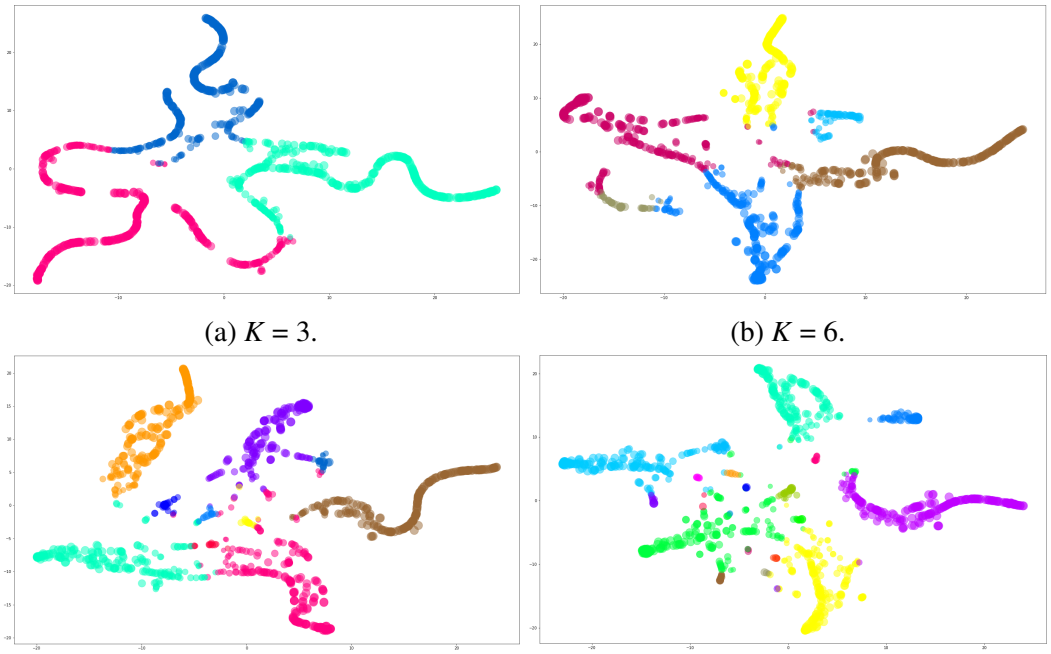



Fig. 4.25 Dataset D1, weighting via TF-IDF. Word cloud representation of a subset of topics for $K = 10$.

topics to describe the corpus, finding a very fine-grained topics model for the dataset.

Comparing the ESCAPE results with the state-of-the-art techniques, which produce as $K$ values 3 and 19 (with RPC and En-LDA respectively), two different scenarios are depicted.
The RPC proposes 3 as optimal number of clusters. This is the same value proposed by the first solution of the ESCAPE framework. As described above, the clustering result is not bad, but some of the original topics are mixed together (*music* and *literature*, *sports* and *mathematics*). In this sense, ESCAPE outperforms RPC giving more options to the analyst, letting her the possibility to choose among different solutions with different granularity levels.

With the En-LDA approach, which proposes 19 as the optimal number of clusters, good partitions are identified (the t-SNE representation of the clustering result is



(a) $K = 3$. (b) $K = 6$.

(c) $K = 10$. (d) $K = 19$.

Fig. 4.26 Dataset D1, weighting via TF-IDF. t-SNE representation, $K$ 3, 6, 10 and 19 respectively.

reported in Figure 4.26d). Indeed, all the original categories of dataset can be recovered in topics. Furthermore, the model identifies very specific topics, that describe only few documents, and often divide the main categories in subtopics, that deal about more specific arguments with respect to main ones. Examples are the *opera* and the *instruments* topics, that both belong to the *music* main category. The modelling is overall good but having more topics that the ones actually required not necessarily means having a better result. Indeed, too many topics may not be useful for the analysis since then the analysts have a more complex result set to consider in their work.

**Qualitative comparison**

For the qualitative comparison, some interesting visualisations are reported to analyse the main topics for each strategy. Figure 4.25 shows a subset of most representative topic descriptions obtained with the TF-IDF weighting schema and $K$ equal to 10. It is easy to interpret and assign a representative label to each topic that actually describes the main categories originally present in the dataset. These topics are also

the bigger ones represented in Figure 4.24, meaning that LDA is able to well identify and split the document into coherent and cohesive clusters. The remaining clusters identified by the LDA model includes words describing more specifically a sub-topic, thus they are not shown in the word cloud representation.

With respect to the computational and time costs, ESCAPE outperforms En-LDA. Calculating En-LDA indices is computationally very expensive, and the number of iterations explodes with the growth in documents vocabulary and the cardinality of the corpus. Furthermore En-LDA needs to be computed for all the topics in the given set, having to find the entropy minimum among all the possible $K$ possibilities. RPC, instead, requires a computational time comparable to the one required by ESCAPE, in the worst case.

Furthermore, with respect to the state-of-the-art techniques, ESCAPE considers the semantic descriptions of the topics to assess the level of separation of the clusters. This is not considered in the state-of-the-art approaches, that only evaluate the goodness of the results by means of probabilistic metrics. In ESCAPE, the quantitative indices of confidence, could be used instead to deeper analyse the proposed results.

### 4.5.2 Informative Dashboard

To analyse in a major detail the partitions obtained by ESCAPE *Informative Dashboards* have been integrated. These visualisations include several graphical representations that are self-explained, which are exploited to simplify and synthesise the extracted knowledge patterns in a compact, human-readable, detailed and exhaustive representation. The proposed visualisation techniques show data and knowledge at different granularity levels. These visualisation techniques allow different stakeholders to easily capture the high-level overview of textual collections through topic detection and document clustering, and drill-down the knowledge to the single document. In this way, people who are not experts or technicians, are able to understand the arguments thanks to the support of these visual representations.

In the informative dashboard, we do not include the technical analysis of the parameters and their comparison with the state-of-the-art. Instead, we analyse the content within the partitions. We have observed that from the analysis of the impact of the weights, different partitions can be represented. In this dashboard the different clusters/topics are analysed to characterise their content.

For each experiment, ESCAPE reports the proposed visualisation techniques, including also some default configurations. In this way, if the end-user is not a domain expert, he could be also able to read the results. Of course, ESCAPE permits to change the default configuration for expert analysts. The proposed informative dashboards exploit different kinds of representations to show data and knowledge at different granularity levels. The proposed visualisation techniques allow different stakeholders to easily capture the high-level overview of topic detected in each corpus.

In an informative dashboard, ESCAPE includes two main aspects which are reported to the analyst: (i) the *topic-term distribution* to understand the main relevant words that characterise each topic, and the *document distribution* to highlight how the documents are related to each other in each corpus. For each aspect, different visualisations have been integrated which are able to show similar information at different granularity levels. In more detail, we have included:

- *Topic-term distribution:* through tabular representation, word-clouds, termite and graph. These techniques report for each topic, the k-top frequent/probable words for each methodology, and are sorted by complexity of understanding. In other words, using the table visualisation, we only report the main relevant words without including the value of probability or frequency of each term. The analyst should only read the words and try to extract a topic. While, for a more detail analysis, the other two representation also include information about the probability/frequency of these k-top words. Specifically, word-clouds represent more probable or frequent words in a bigger size. This format is useful for quickly perceiving the most prominent terms. Termite visualisation includes also a major level of detail, which is the simple way of comparison between different topic. As a matter of fact, the belonging of each term is represented as a point in the plot, whose size depend on the probability/frequency that that term should be taken from the topic during the creation of the document. However, since the words are independent from the topic, the analyst is able to easily compare if same words are used for more topics. Moreover, looking at the size of each point, the analyst is able to characterise each cluster in more detail. Lastly, the graph visualisation includes a lot of interesting aspect; it represents a complex visualisation, but that shows a lot of information and is also readable by non-experts. The structure itself

is complex, however is really informative. Indeed, the analysts are able to navigate the graph and see how the words are connected together, analysing the impact of each edge which represent the frequency/probability of terms, and analyse more words which using other techniques could be more complex. Although the creation of the graph is complex, its usefulness is high.

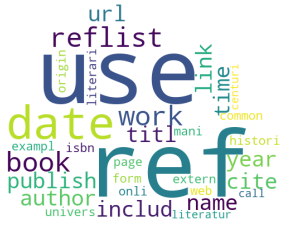
- *Document distribution:* through correlation matrices and t-SNE representation. These techniques are able to show different information for each dataset, since the correlation matrix is able to analyse the dependency of documents in the entire corpus, not focusing only on the most probable and frequent words, and the t-SNE is able to show how the points are displays into a lower space using a non-linear dimensionality transformation. In this way, the analyst is able to analyse the relationships between the data points (i.e. documents) at different level of detail; specifically, the correlation matrix including all the possible couples, while the t-SNE trough the entire corpus.

We recall that in the technical dashboard we have reported the two highest similarity weighting schemas, which are the TF-IDF and the Boolean for both the topic modelling approaches. Of course, the partitions are not the same because of the ARI index tends to 0.554 and 0.790, respectively. However, analysing only the values is not sufficient to quantify the similarity between the topic. Below, we reported the analysis of these two weighting strategies to highlight the main differences between the two approaches.

### TF-IDF weight

Here, we analyse the impact of the TF-IDF weighting function on both the methodologies integrated in ESCAPE. To this aim, we reported the word-cloud comparison for the weighting schema TF-IDF for both the methodology. Specifically, in Figure 4.27 and in Figure 4.28 are reported the 10 word-clouds related to the joint-approach, while in Figure 4.29 and in Figure 4.30 are reported the 10 ones related to the LDA modelling. Analysing the main probable words for each topic, we can extract the fallowing considerations.

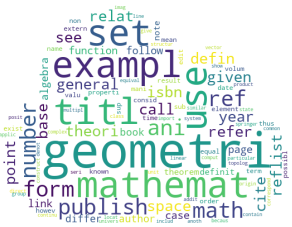
In both the partitions found, we have 10 clusters. However, the partitions should not be the same, since the value of ARI index is not 1. Moreover, we recall that the 5




(a) Cluster0


(b) Cluster1


(c) Cluster2



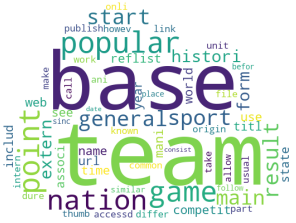

(d) Cluster3


(e) Cluster4.

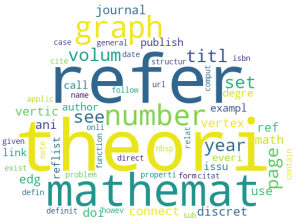
Fig. 4.27 Dataset D1, weighting via TF-IDF. Word-cloud representation from cluster 0 to 4 for the Joint approach.

a-priori known categories are: *cooking*, *literature*, *mathematics*, *music* and *sport*. We aspect to find these themes in these 10 partitions.

Firstly, we report a summary of the found topic in Table 4.60. We can highlight that although the partitions are equivalent in number (10 topics), the meaning of the found topics are different. In fact, the five macro categories are correctly identified by both approaches, however the algebraic method finds subdivisions for the mathematics

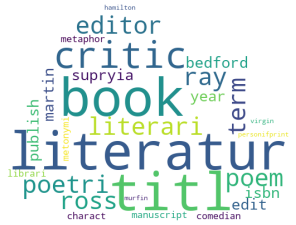



(a) Cluster5




(b) Cluster6


(c) Cluster7



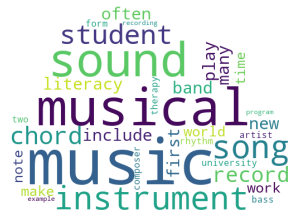

(d) Cluster8


(e) Cluster9

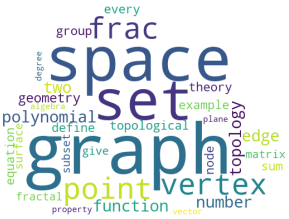
Fig. 4.28 Dataset D1, weighting via TF-IDF. Word-cloud representation from cluster 5 to 9 for the Joint approach.

and sport categories, while the probabilistic method for literature and sports. The results are both satisfactory.

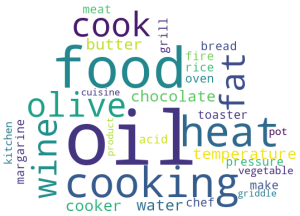
We also include the analysis of the correlation between the founded partitions. For the joint approach we report the correlation matrix in terms of hot-cold topic. In this way, the colors help the analyst to read the possible correlation between the topics. We use the red color to high correlation between partitions (see Figure 4.31). On the



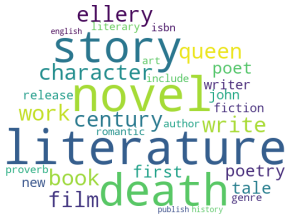

(a) Cluster0




(b) Cluster1




(c) Cluster2



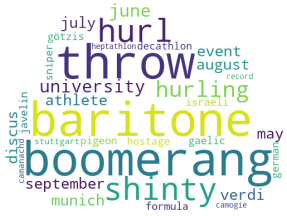

(d) Cluster3


(e) Cluster4

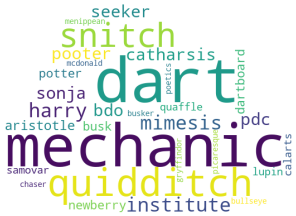
Fig. 4.29 Dataset D1, weighting via TF-IDF. Word-cloud representation from cluster 0 to 4 for the Probabilistic approach.

other side, in the probabilistic approach we report the graph representation, able to help the end-user to analyse the possible intersection between words in the different topics (see Figure 4.32). To compute the correlation matrix, ESCAPE first sorts the clusters based on their cardinality, then calculate the correlation between all the pairs of documents.

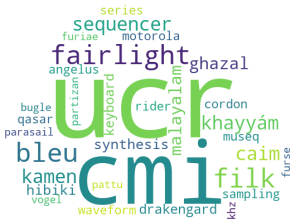

(a) Cluster5



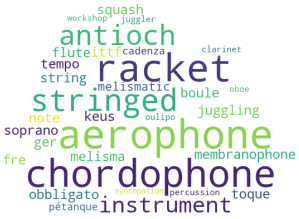

(b) Cluster6




(c) Cluster7




(d) Cluster8




(e) Cluster9

Fig. 4.30 Dataset D1, weighting via TF-IDF. Word-cloud representation from cluster 5 to 9 for the Probabilistic approach.

From Figure 4.31, we can notice a high correlation between cluster 4 and 5, which analysing Table 4.60, (column Topic Joint-Approach) are both related to sports. Moreover, there is another correlation between 3 and 6, which looking always at Table 4.60 or also the previously presented word-clouds, are both related to maths topics. Specifically, cluster 3 is related to several maths topics, while cluster 6 is inherent mainly to graph theory.

| ClusterID | Topic Joint-Approach | Topic probabilistic Modelling |
|---|---|---|
| Cluster0 | Literature | Music |
| Cluster1 | Food | Maths |
| Cluster2 | Music | Oil Food |
| Cluster3 | Maths | Literature |
| Cluster4 | Sport | Sport |
| Cluster5 | Sport | Dynamic sport |
| Cluster6 | Graph Theory | Music |
| Cluster7 | Music | Quiddich - Literature |
| Cluster8 | Literature | Literature |
| Cluster9 | Oil | Musical Instruments |

Table 4.60 Topic description for dataset D1 for both the approaches.

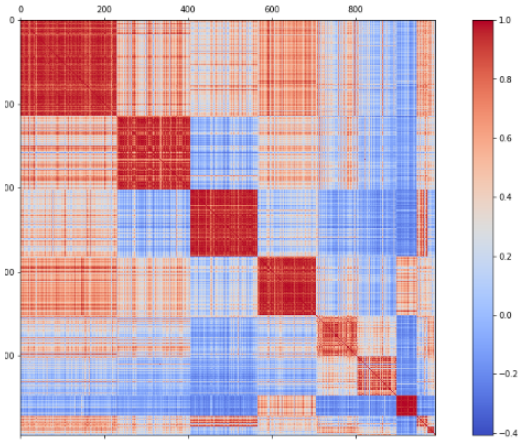



Fig. 4.31 Dataset D1, weighting via TF-IDF. Hot-topic correlation matrix representation, TF-IDF weighting schema, $K$ 10, joint approach.

On the other side, Figure 4.32 reports the graph representation for the probabilistic LDA modelling. The most relevant words for each topic, (i.e., the words which are most likely to belong to a particular topic) are well-separated, as can be deduced from the graph analysis. Considering both the top-20 (see Figure 4.32 (Left)) and the top-40 (see Figure 4.32 (Right)) words, the graph is still very disconnected, indicating that the analysed partitions are well separated.

Another way to compare the found partitions with respect to the two approaches is the analysis of the t-SNE representations, which give the analyst the possibility to plot into a lower space (i.e., 2D in our framework) the high dimensional data under analysis. This representation is reported in Figure 4.38. We remind that

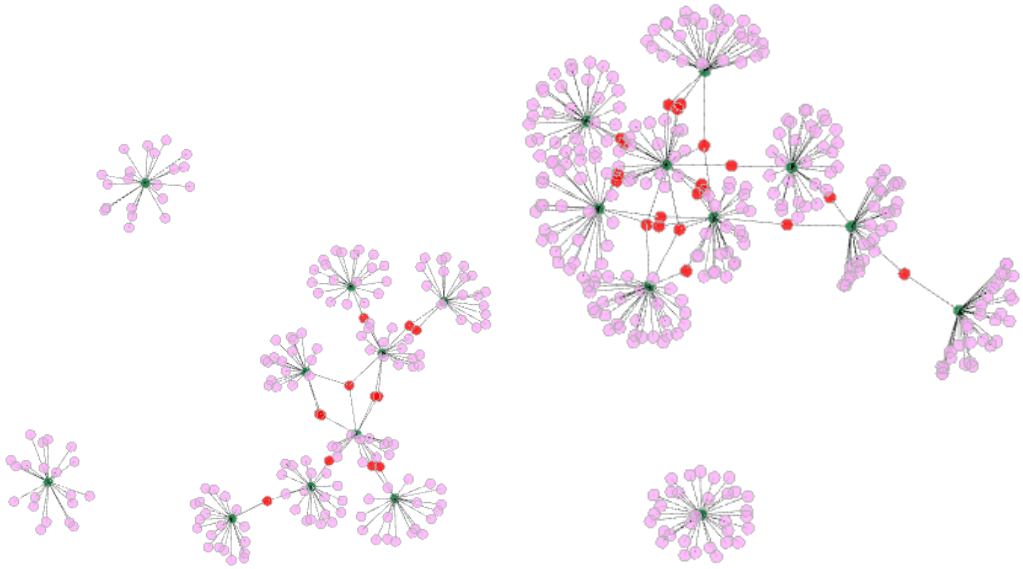


Fig. 4.32 Dataset D1, weighting via TF-IDF. Graph representation. Top-20 (Left) and top-40 (Right) words, $K$ 10 for the Probabilistic approach.

the t-distributed Stochastic Neighbour Embedding (t-SNE) is a machine learning algorithm for visualisation, which is based on a non-linear dimensionality reduction technique well-suited for embedding high-dimensional data for visualisation in a low-dimensional space. It is based on the concept of probability distribution, indeed it constructs a probability distribution over pairs of high-dimensional objects in such a way that similar objects have a high probability of being picked, whilst dissimilar points have an extremely small probability of being picked.

A key feature aspect of t-SNE is a tunable parameter, *perplexity*, which we have presented as quality metric to evaluate the goodness of the probabilistic LDA modelling. This parameter says how to balance attention between local and global aspects of the data under analysis. The parameter is related to the concept of the number of close neighbours each point has. The perplexity value has a complex effect on the resulting pictures, indeed since the algebraic model is not born to measure the perplexity in probabilistic terms, the good value to be set for its plot could be complex. While for the LDA model there is no problem, we use the value returned by ESCAPE. In Figure 4.38 we reported the two representation of the t-SNE visualisation for the joint approach (Top) and for the probabilistic approach (Bottom). The shape is quite similar, however the plot using the LDA model converges better in the presented

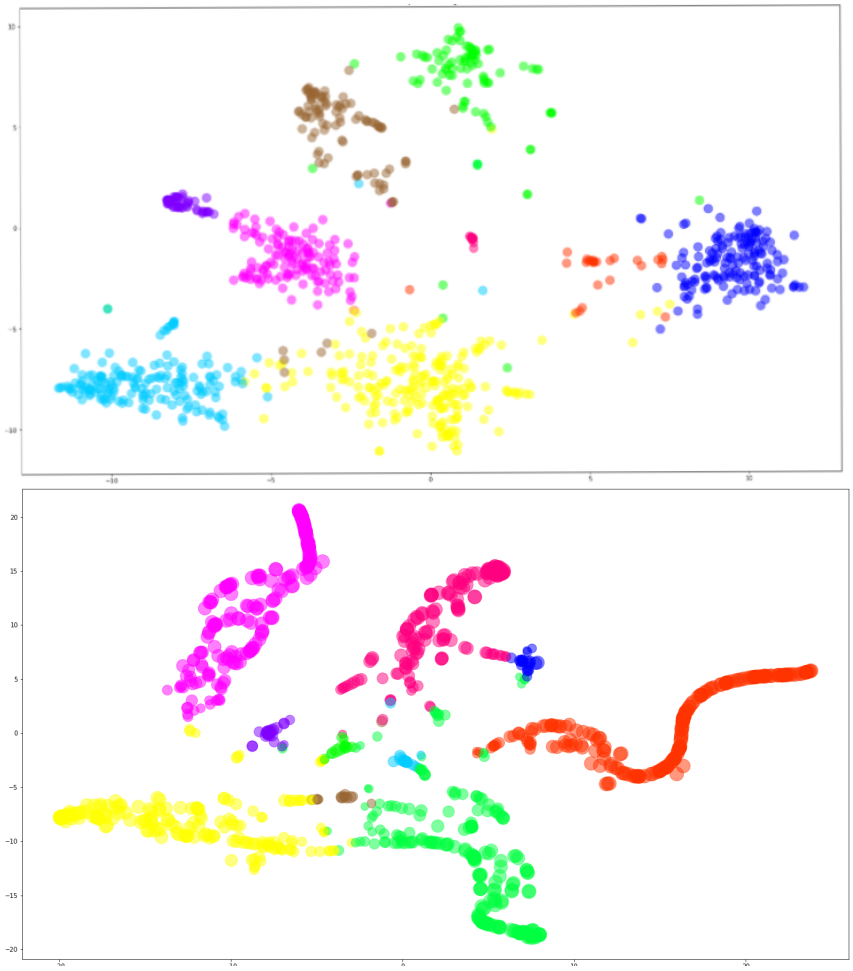


Fig. 4.33 Dataset D1, weighting via TF-IDF. t-SNE representation, with $K$ 10. Joint-approach (Top) and Probabilistic approach (Bottom).

figures. Probably, it is bad news that to see global geometry shape it is necessary a fine-tuning perplexity parameter. Moreover, since real data are characterised by multiple clusters with different cardinality (i.e., number of documents), it could happen that using only one single perplexity value is not enough to capture distances across all clusters. Indeed, the perplexity metric is a global parameter defined for the entire model. Thus, an interesting area for future researches could be the fixing of this problem.

### Boolean weight

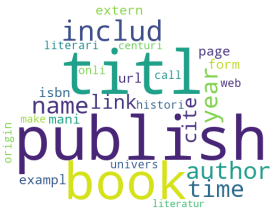
The highest value analysing the ARI between the two approaches for dataset D1 is computed for the Boolean weighting strategy. It highlights a great similarity between the two partitions. Moreover, the number of documents in each cluster is comparable. In the joint-approach we have integrated two weighting strategies with respect to the local weight Boolean, which are Boolean-IDF and Boolean-Entropy. However, since the two partitions are really similar as shown in subsection 4.3.3, we only consider the Boolean-IDF as comparison with respect to the Boolean-TF used for the LDA modelling.

We report in Figure 4.34 and Figure 4.36 the word-clouds of the two approaches, respectively. Specifically, Figure 4.34 is related to the five-topic found using the algebraic approach, while Figure 4.36 is related to the probabilistic model. In detail, analysing Figure 4.34, we can observe that with respect to the TF-IDF local weight, the analysis is less precise. We can extract the main topic from each word-cloud; however, the partitions present more common words used for more topic.

Another useful way to identify the latent topic, is the *termite representation*. Termite exploits a tabular layout visualisation to promote the comparison of distinct words both within and across the latent topics, based on the co-occurring terms. The Termite representation of the clustering obtained with $K$ equal to 5 is reported in Figure 4.35. The represented topics are however well described, and they identify the original categories of the dataset.

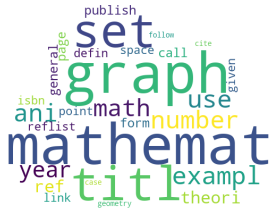
For the probabilistic model, we can highlight that when we considering the clustering obtained with $K$ equal to 5 and its topic descriptions, when looking at the word clouds in Figure 4.36, many terms (such as *include* or *first*) appear to be in all the groups of the most significant words for each cluster. This happens because the Boolean-TF weighting schema give more relevance to words which appear most in the whole corpus, without penalise them. However, it could lead that these words do not belong to any specific topic, or they just do not bring any additional information useful for the topic modelling description phase. To this aim, we have included a post-processing phase for this particular weighting schemas.

In order to not consider these terms and bring up the words characterising the topics identified by the LDA modelling process, we apply a further post-processing step to evaluate the results. Once the models have been created and the $K$ values selected,




(a) Cluster0


(b) Cluster1




(c) Cluster2


(d) Cluster3


(e) Cluster4

Fig. 4.34 Dataset D1, weighting via Boolean-IDF. Word-cloud representation, with $K$ 5, for joint approach.

we took into consideration more words to describe the topics, and then we remove from them all the words appearing at least in four topic representations.

The results obtained by this post-processing operation are reported in Table 4.61. By this way, the most common words not bringing specific information has been excluded from the descriptions, and the terms relevant for the meaning of the categories are visible to the analysts. As a matter of fact, the assigned labels to the clusters generated by the LDA model cover the following main topics: *sport*, *math-*

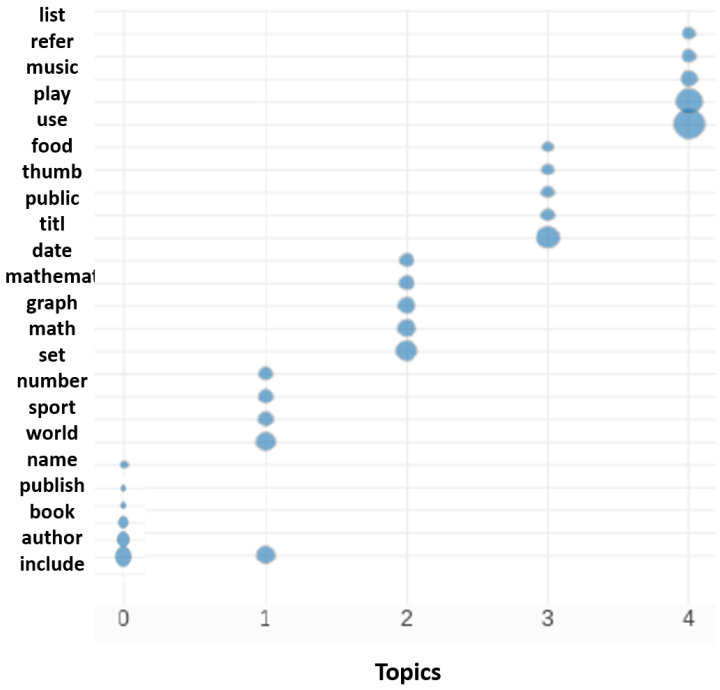



Fig. 4.35 Dataset D1, weighting via Boolean-IDF. Termite representation, with $K$ 5, for joint approach.

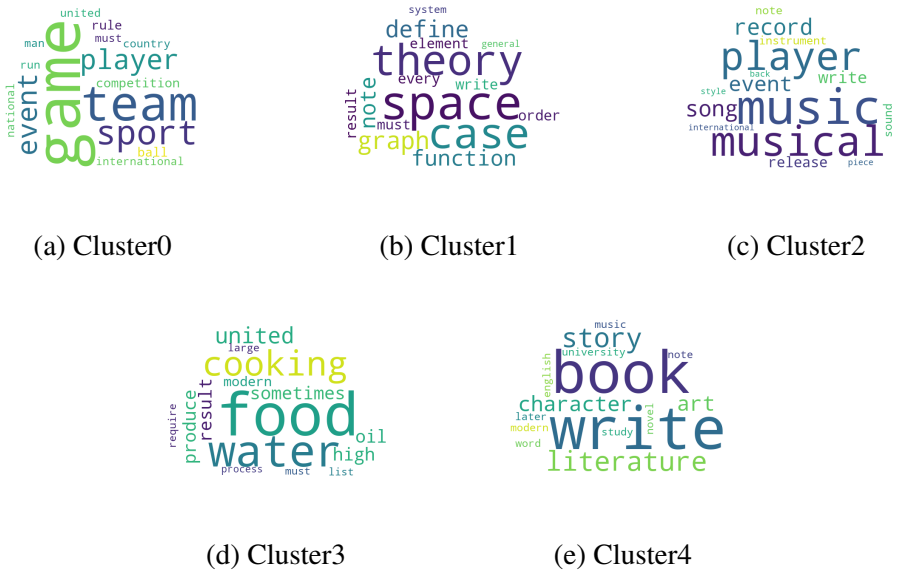

(a) Cluster0 (b) Cluster1 (c) Cluster2

(d) Cluster3 (e) Cluster4

Fig. 4.36 Dataset D1, weighting via Boolean-TF. Word-cloud representation, with $K$ 5, for probabilistic approach.

| *K* | Topic description |
|---|---|
| **1** | game, team, sport, player, event, competition, ball, rule, international, must, country, united, man, national, run |
| **2** | space, theory, case, graph, define, function, note, every, write, order, result, element, must, system, general |
| **3** | music, musical, player, record, song, event, write, release, instrument, note, sound, international, style, piece, back |
| **4** | food, water, cooking, united, sometimes, produce, result, high, oil, modern, large, require, must, list, process |
| **5** | write, book, literature, story, character, art, university, music, novel, modern, english, word, note, study, later |

Table 4.61 Dataset D1, weighting via Boolean-TF. Topic-terms representation, with *K* 5, probabilistic approach.




(a) Cluster0 (b) Cluster1 (c) Cluster2

(d) Cluster3 (e) Cluster4

Fig. 4.37 Dataset D1, weighting via Boolean-TF. Word-cloud representation,with *K* 5, probabilistic approach after post-processing.

*ematics*, *music*, *cooking*, *literature*. Using this post-processing approach, we are able to describe perfectly the macro-categories of this dataset, as shown in the new word-clouds reported in Figure 4.37.

To better show the impact of removing words that appears at least in four topics, we report the graph representation before and after this improvement. Figure 4.38 shows the graph representation analysing the top-20 words for each topic. Specifically,

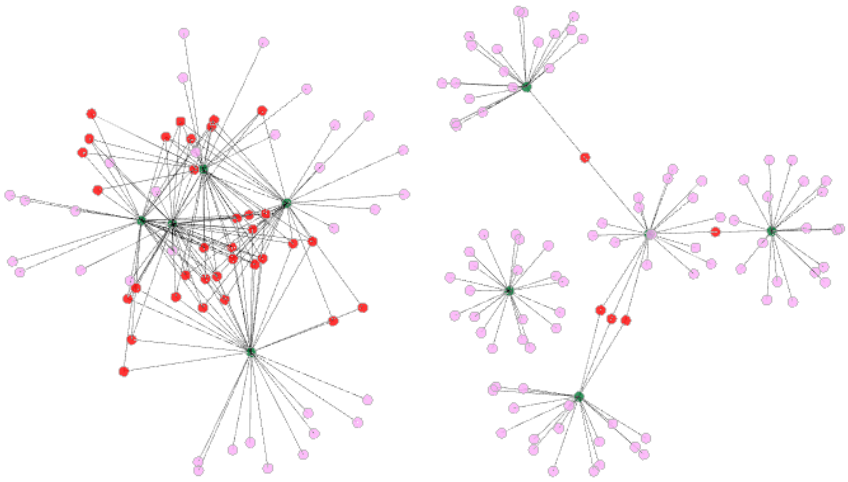


Fig. 4.38 Dataset D1, weighting via Boolean-TF. Graph representation, with $K$ 5, without post-processing (Left) and with post-processing (Right).

on the left, is reported the case without the post-processing, while on the right, we reported the case with the proposed post-processing. The first graph is more connected with respect to the second one; moreover, form the analysis of the graph after the post-processing, we can see the improvement of this phase, since the new graph is not connected at all. This means that the words that describe each topic are well-separated from cluster to cluster.

## 4.6 ESCAPE final considerations

From the analysis of the obtained experimental results, we can assess that ESCAPE well performs in describing the seven corpora under analysis, clustering the documents based on their main content. The proposed framework is able to group the documents into well separated topics. Both the exploratory methodologies are able to split the corpora into well separated groups, both in terms of quality indices and easily-interpretable graphical representations.

We have observed that the joint approach, which is based on a dimensionality algebraic phase before the application of the partitional K-Means algorithms, is able to find homogeneous partitions in term of documents for each cluster. In other words, this approach creates balanced clusters. Moreover, changing the weighting strategy, the end-user is able to clusterise the same dataset, at different granularity levels.

Specifically, we have learnt that the local weight LogTF tends to find a small number of clusters, as shown in Section 4.3. Moreover, the weighting schemas TF-IDF and TF-Entropy have a high value of ARI for all the textual corpora; indeed, these two weighting schemas are able to characterise at a very fine level of detail the corpus. We have also seen that the global weight IDF is able to create more clusters able to find also sub-topics related to the major category. In other words, this global weighting schema is able to characterise each dataset in a more precise way. While, on the other hand, the Entropy is able to find less clusters but with a larger cardinality, finding only the main relevant topic associated with each partition.

On the other hand, the weighting schemas that most differentiate the same dataset are the TF-Entropy and Boolean-IDF. As a matter of fact, since the partition's cardinality is completely different, the value of the ARI tends to decrease, i.e., the partition are different in terms of clusters. However, in all the corpora the ARI index never tends to zero, since at least the main topics are discovered. In fact, it is true that the weighting schemas describe the topics at a different granularity level, however, when sub-topics are highlighted, there are particular and detailed topics of one of the main topics. In detail, considering both a more coarse level and a finer level, macro-arguments are always identified, independently by the weighting schema used. The difference is in the smaller clusters which characterise the macro-topics. Therefore, ARI almost never reaches levels that are too low, precisely because the clusters are well balanced and the macro categories are identified in all the experiments.

On the other side, for the probabilistic approach, considering the semantic similarity among the produced topics turned out to outperform the current used approach to find proper numbers of clusters. Indeed, the proposed algorithm is able to capture the effective cohesion level of the clusters, and then properly identify the optimal number of topics. The results obtained from all the datasets considered in the thesis confirm the clusters are well separated, especially for certain weighting schemas such as TF-IDF. However, with respect to the joint-approach, some weighting schemas lead to very poor results, such as the Entropy-based schemas. In general, the probabilistic model tends to find more heterogeneous clusters.

Analysing all the ARI comparisons for each corpus, an analyst can observe that the LogTF-Entropy weight is the one with lower values when we compare its partition with respect to the other partitions. Moreover, analysing the respectively cardinalities of the partitions, we can highlight that the results obtained using this weighting

schema are the most inhomogeneous. There is always a very large cluster, which includes more than 80% of documents. Moreover, LogTF-Entropy allows also the generation of empty or singleton clusters when we apply LDA modelling (i.e., clusters with only one document). Of course, this kind of partition is not useful for the analyst; as a matter of fact it is like to categorise all the documents in a single macro topic, which represents the union of all the other categories. Also, for the other weighting schemas associated with the global Entropy weight, we can highlight that the number of partitions include a larger cluster with respect to the others, even if smaller than the one created by the LogTF-Entropy. However, despite these schemas, the other results are also satisfactory

On the other hand, the IDF global weight is the one which is able to find a number of clusters higher than the expected value for the probabilistic modelling. Indeed, from the analysis of both the cardinalities and the corresponding ARI, we can observe that in mean the value of the ARI is the highest for the two weighting strategies associated with this global weight.

ESCAPE turns out to be really helpful for the analysts during the knowledge discovery process. Indeed, the analyst can choose to assign to the words in the documents different relevance by means of different weights and compare the solutions obtained using the two approaches, analysing the different granularity levels. The best partitions can also be compared using innovative visualisation techniques, able to help the analyst during the validation step. Moreover, the two proposed dashboards are able to characterise different aspects in which the analyst is interested to, including also the possibility of comparing the proposed approaches with respect to the other state-of-the-art techniques.

# Chapter 5

# Conclusion and Future Work

This document has presented the main activity carried out during my Ph.D. studies. In these three years, I have been able to design and develop a new framework, named ESCAPE (**E**nhanced **S**elf-tuning **C**haracterisation of document collections **A**fter **P**arameter **E**valuation), able to support the analyst during all the phases of the analysis process tailored to textual data. ESCAPE includes three main building blocks to streamline the analytics process and to derive high-quality information in terms of well-separated and well-cohesive groups of documents characterising the main topics in a given corpus.

Firstly, the data distribution of each corpus is characterised by several statistical indices (e.g. Guiraud Index, TTR). The jointly analysis of these statistical features is able to describe the lexical richness and characterise the data distribution of each collection under analysis. Then, a pre-processing phase is applied to prepare the textual content of documents for the next phases. These activities, which are done subsequently, represents each document via the Bag-Of-Word (BOW) representation. Using this model, a text (e.g. a sentence or a document) is represented as the bag (multi-set) of its words, disregarding grammar and even word order but keeping multiplicity. To measure the relevance of these multiplicities, ESCAPE includes several weighting strategies, which are able to measure term relevance in the same dataset by exploiting a local weighting schema (e.g. TF, LogTF) together with a weighting schema (e.g. Entropy, IDF). ESCAPE automatically exploits all the possible combinations of local and global weighting schemas to suggest to the user the ones that well model the term relevance in the collection under analysis. Since

we are interested in finding out the number of topics contained in a given collection of documents, in ESCAPE we have integrated two strategies because no strategy is universally superior. Specifically, we have integrated:

- an algebraic model based on SVD decomposition together with the K-Means clustering algorithm (i.e., the joint-approach);
- a probabilistic model, based on the analysis of latent variables through the LDA (i.e., the probabilistic method).

Each strategy has been enriched with a self-tuning methodology to automatically set the specific-input parameters required by each involved algorithm. This allows off-load the end-user to correctly configure input parameters that is usually a time consuming activity. Lastly, several user-friendly and exhaustive informative dashboards have been embedded to help the end-user to effectively and efficiently explore the results. To evaluate the quality of corpora partitions automatically discovered by ESCAPE, a variety of quality indices have been integrated into the proposed framework.

We have tested ESCAPE on different real textual corpora characterised by different data distributions. Selected collections include documents that are both dense and rich in the number of distinct terms, with medium-long document lengths, and short documents that are characterised by a variable lexical richness and a limited vocabulary.

Performed experiments demonstrate the effectiveness of ESCAPE in identifying good partitions of a given corpus modelling its variety of different latent topic. The proposed algorithms and the visualisation methods have made the analyst's task in testing the different experiments more comprehensible and simplified. Based on the experience acquired during the deeper analysis of the experimental results, the following observations have been learnt:

- Both the exploratory methodologies are able to split the corpora into *well separated groups* of similar documents, both in terms of quality indices and easily-interpretable graphical representations.
- The different weighting strategies are able to characterise the same dataset at *different granularity level* for both the methodologies. This means that by

changing the weighting schema the analyst is able to exploit the corpus based on the level of detail required.

- The joint-approach is able to find always balanced partitions, due to the properties of the clustering algorithm used (i.e. K-Means).
- The probabilistic approach is much more influenced by the weighting strategy, and some weighting schemas should be skipped by the analyst since could lead to very poor results (e.g. Entropy-based schemas). In general probabilist model tends to find heterogeneous clusters.
- Even if the number of topics found is similar for both methods, the partitions found model different aspects of the collection under analysis. Thus, ESCAPE includes two different approaches both providing very interest and useful results, although sometimes different.

Although the results obtained in these years are satisfactory, different directions have yet to be analysed and explored. Possible future extensions concern the *integration* in ESCAPE of:

1. *New data analytics algorithms* to exploit other interesting models. Specifically, we are currently including:
    - other *algebraic data reduction algorithms* (e.g. Principal Component Analysis (PCA) [176]) for the joint-approach together with the exploitation of other clustering methods (e.g. hierarchical algorithm) and other *probabilistic topic modelling methods* (e.g. Probabilistic Latent Semantic Analysis (pLSA) [43]);
    - *autoencoder-based data reduction algorithms* to compress the information of the input variables into a reduced dimensional space and then recreate the input data set;
    - *non-parametric models* (such as Deep Neural Network DNNs [177] and K-NNs [178]) to promote the comparison with other state-of-the-art techniques;
    - more *weighting functions* (e.g. aug-norm) to underline the relevance of specific terms in the collection;

- more *statistical indices* to characterise the corpora distribution.

2. A *semantic component*: (e.g. WordNet [179]) able to support the analyst in a double phase. Both during the pre-processing phase, to eliminate semantically bound words, in this way we are able to reduce the dictionary and also the complexity of the algorithms, and also during the post-processing phase. In this way, we could analyse through the most relevant words for each topic, those that are related to each other, helping the analyst in understanding the outputs. Specifically, each topic can be characterised by words which are semantically related, and so could represent subtopic of the same macro category. Moreover, thanks to the ontological base, the analyst could also add a hierarchy level for each word of the dictionary to support other analytics tasks (e.g. generalised association rules discovery).

3. A *Knowledge-Base*: to store all the results of the experiments, including the data characterisation and the top-k selected results, for each methodology and weighting schema to efficiently support self-tuning methodologies.

4. A *self-learning methodology*: based on a classification algorithm trained on the knowledge base content to forecast the best methods for future analyses. So, when a new collection need to be analysed, ESCAPE should compute the data distribution characterisation through statistical features and suggest possible good configurations without performing all the analytics tasks.

We are currently working to release ESCAPE shortly, to allow users to really exploit our system, easily perform interesting analysis and get useful knowledge from their data, also without detailed knowledge about the integrated algorithms or without wasting time to configure the parameters. Based on user experiences, we could test our framework on other interesting and different datasets. In this way, we would be able to understand any drawbacks and improve the features of ESCAPE. Moreover, getting real feedbacks from people, we would also be able to improve the current implementation.

The research proposed in these three years has produced different results but leaves a great open door to improve more and more. Only a subset of the research activity described in this dissertation has been already published. Specifically, some preliminary ideas of the engine presented in this work are described in [3, 7]. The

joint-approach has been discussed in [5] and its tailored version to Twitter collections has been presented in [6]. The probabilistic model has been described in [4], while a first design of the informative dashboard has been presented in [180].

# Appendix A

# Self-tuning strategies in other domains

This Appendix contains a set of experiments and results that investigate further the application of self-tuning strategies and automatic setting of algorithm parameter values in other domains as reported in Section 1. Specifically, different methodologies have been implemented and developed to help the end-user in setting the algorithm's parameters without knowing the complexity of each algorithm. Specifically, two different types of data have been analyed: (i) structured data and (ii) stream of data. For the structured data (see Section A.1), a detail analysis is done on the analysis of collections of energy-related data enriched with meteorological features; while for the stream of data (see Section A.2), a methodology for determining suitable groups of residential consumers is addressed, based on time series of their hourly energy consumption and contractual data.

An ever increasing quantity of energy data is produced every day in our lives and society. These data are continuously collected through several smart meters from different smart-city environments. The analysis of energy-related data collections has received increasing consideration from different and cross-research communities, including energy, data mining, databases and statistics communities. These data collections have great potential because an interesting subset of high value knowledge (e.g., detailed patterns and models to characterise energy consumption at different granularity levels) can be detected to support several stakeholders during the process

of decision-making (e.g., energy managers, energy analysts, consumers, building occupants).

Extracted knowledge items have a great potential to influence the overall energy balance of our communities, in particular by optimising the building thermal energy consumption, which mainly consists of (i) a static contribution, that is determined by the building structure (e.g., walls, windows, materials, captured by the building energy signature) and appliance energy ratings, and (ii) a dynamic component, that is provided by the usage behaviors and the lifestyle of the people living inside the buildings. With the aim of reducing energy demand, people should be more aware about their building consumption to pursue energy-saving actions. Innovative analytics methodology should be devised to provide interesting and actionable knowledge items about energy consumption in buildings. The knowledge items should be easily interpretable by people to be effectively exploitable.

Furthermore, the influence of multi-dimensional weather data on energy consumption has been condensed into few attributes (e.g., the temperature and humidity) in most existing approaches, due to the complex nature of the full set of meteorological conditions, and the difficulty of automatically identifying the most relevant correlations with many variables. Hence the need to address such correlations with self-learning transparent approaches, which harness the power of complex algorithms to the benefit of energy-domain experts and citisens.

## A.1 Structured data

To the analysis of this kind of data, a data mining engine has been proposed, named METATECH (METeorological data Analysis for Thermal Energy CHaracterisation) [8], covering the whole analytics work-flow of energy-related data. METATECH analyses collections of energy-related data enriched with meteorological features through a two-fold methodology based on cluster analysis and generalised association rules to automatically extract and transparently describe energy consumption patterns correlated with meteorological data.

The joint approach based on both cluster analysis and generalised association rules allows an efficient characterisation of the energy consumption. Specifically, the clustering analysis targets the unsupervised discovery of groups of different thermal

energy consumption that occurred with similar weather conditions. Each cluster is then characterised by an ordered list of interesting patterns at different granularity levels, to summarise the cluster content and to highlight interesting correlations among thermal energy consumption and meteorological conditions. METATECH exploits the K-means algorithm to cluster weather data, jointly with a self-tuning strategy to automatically discover the desired number of groups, while the generalised association rule miner extracts correlations among energy data and meteorological conditions. A categorisation of rules into reference classes, based on meaning, is proposed to ease the manual inspection of the results and their understanding. The model of the data is transparent as it consists of rules, in the form of correlations among different attribute values, which are directly readable by humans. The full process is designed to self-learn from the data how to proceed at each step, by tuning parameters, partitioning the data, and identifying the most relevant rules among the full set of correlations that exist in the data.

Extracted knowledge items can support energy managers in the decision-making process, for example through the definition of proper strategies to efficiently satisfy the energy demand for different buildings. Furthermore, extracted knowledge items can enhance people's (consumers and building occupants) awareness of energy consumption and plan ad-hoc strategies to reduce the building consumption during some critical time slots (e.g., energy peak demand) or when rooms are empty.

As a case study, METATECH has been validated on real energy consumption data collected in a real-world system available in a major Italian city. These data have been integrated with meteorological information available through a web service.

Experimental results show that the proposed approach is effective in discovering interesting correlations to raise people's awareness of their energy consumption.

The main novelties of METATECH are twofold. (1) It is a self-learning joint approach, based on both cluster analysis and generalised association rules, able to automatically extract interesting knowledge patterns and make them easily interpretable to characterise thermal energy consumption. In particular, the model self-learns, i.e., automatically infers from data the patterns and their correlations, without prior knowledge and with limited user interaction, thanks also to the automatic tuning strategies of the algorithm parameters. (2) It analyses real-world data collected in a heating system available in a major Italian city and presents experimental results of interest for domain experts.

## A.2 Stream of data

The growing availability of data gathered from smart meters requires appropriate procedures for extracting useful information from a continuous flow of metered data. Clustering techniques can be used for the analysis of a large amount of data (e.g., coming from the nation-wide installed meters). However, a key aspect of the analysis is the definition of the features to be used as inputs to the clustering procedure. This definition is not unique and depends on the type of consumers analysed (e.g., residential, industrial), the time step of the data available (typically 15, 30 or 60 minutes), and other information that may be available from the company's databases. Among the various types of consumers, individual residential consumers are the most challenging ones to be addressed, as their consumption depends on a number of behavioural aspects conditioning the regular or irregular way to use the appliances during time, as well as other factors such as number of inhabitants, net income, age of the persons in the family, employment status, and other socio-demographic information.

The variability of the energy consumption patterns for residential consumers requires specific formulations of the input data for clustering, as classical assumptions such as the use of time series together with the Euclidean distance metric are rather inappropriate. In fact, for example the use of the Euclidean distance leads to high distances if two patterns contain a base load and a similar peak located in different time periods in the two patterns. However, this diversity among the positions of the peaks could be only due to the non-regular usage of the same appliance during the day and appears even for two patterns of the same consumer. To avoid the effect of such diversity on the results of load pattern grouping, it is possible to use specific metrics such as dynamic time warping, which tends to create an optimal alignment between the patterns by stretching the horisontal scale.

To this aim, a comparison among different ways to represent the data of residential consumers for creating consistent consumer groups through cluster analysis have been presented in [11]. The available data include the time series of household consumption and their contractual power. Neither categorical data nor socio-demographic information is available. As a real case study, is it illustrated how to effectively analyse the contractual data and the time series of the hourly energy consumption gathered for 10,000 consumers for one year. The importance of data normalisation is discussed by providing specific examples. The effectiveness of using

shape-based representations constructed by using the time series data (from conventional regular patterns to the application of dynamic time warping) and by applying the methodology named CONDUCTS (CONsumption DUration Curve Time Series) developed in [12] is assessed and compared. In addition to data normalisation issues, the contents presented include, the definition of the type and number of features to be used in the analysis, the choice of the number of clusters, the execution of the clustering procedures, the evaluation of clustering validity indicators that express the quality of the clustering results.

The residential consumers may have different consumption patterns depending on the period of the year (e.g., winter, summer), in which some appliances used could be different, and the lifestyle of the consumers could change. As such, mixing the data for the whole year is not effective. Furthermore, some differences could arise between weekdays and weekend periods. On these bases, a suitable way is to identify sub-periods (e.g., months or other sub-partitioning). This paper exploits the concept of creating time windows of user-defined duration, in which the consumption patterns belonging to the same time window are handled together. This approach requires the definition of the time window parameter, which determines the temporal context of interest for the analysis. The time windows can be of different length depending on the period analysed. More specifically, if the time window is very short, only the most recent consumption of the customer will be analysed, but similar behaviours could appear in adjacent time windows and many similar cases could be generated in the study. Instead, a too large time window allows analysing many data on past customer electricity consumption, but it may introduce noisy information in the cluster analysis. In this paper, the value for the time window has been set to two weeks for the weekdays and to one month for the weekend days. Furthermore, the presence of special days such as religious holidays at fixed dates, bank holidays, and local festivities, has been considered by including these special days in the category of weekend days.

Two types of features are considered:

1. The time series of the hourly data generated by the electricity meters (stream analysis).

2. A selected set of features originated by the normalised duration curves constructed by using the hourly data metered in the same time period.

The rationale of using this type of features is of specific relevance for the residential consumption patterns. In fact, in the type of analysis considered in this paper, the residential consumers that use the same appliances in different periods of the day can be considered as similar. This happens because the focus of this paper is to categorise the consumers from the whole pattern and not to address specific timing issues relevant to the formulation of demand response programmes. Let us consider a two-week period with 240 hourly data from weekdays. These data are first ordered in the descending order to construct the duration curve. Then the variations determined from two successive data on the duration curve are considered for each consumer, and the cumulative distribution function (CDF) of the average variations for all consumers is calculated at each point of the duration curve. The resulting CDF has 239 points (because of the calculation of the variations, the first point is excluded). This CDF is then cut into a given number of proportional intervals (e.g., deciles) on the vertical axis, and the corresponding cut points are selected. In this paper, nine cut points have been selected, by excluding the first and the last point of the deciles limits. A key aspect is that the definition of the cut points is done by considering the whole set of consumers in the time window. After that, the same cut points are used to pick up the nine values referring to each consumer. These nine points are the selected features that are assumed to describe the behaviour of each consumer.

# References


[1] Pinar Alper, Khalid Belhajjame, Carole Goble, and Pinar Karagoz. Small is beautiful: Summarizing scientific workflows using semantic annotations. In *Big Data (BigData Congress), 2013 IEEE International Congress on*, pages 318–325. IEEE, 2013.

[2] Ciprian-Octavian Truică, Jérôme Darmont, and Julien Velcin. A scalable document-based architecture for text analysis. In *International Conference on Advanced Data Mining and Applications*, pages 481–494. Springer, 2016.

[3] Tania Cerquitelli, Evelina Di Corso, Francesco Ventura, and Silvia Chiusano. Data miners' little helper: data transformation activity cues for cluster analysis on document collections. In *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics*, page 27. ACM, 2017.

[4] Stefano Proto, Evelina Di Corso, Francesco Ventura, and Tania Cerquitelli. Useful topic: Self-tuning strategies to enhance latent dirichlet allocation. In *2018 IEEE International Congress on Big Data (BigData Congress)*, pages 33–40. IEEE, 2018.

[5] Evelina Di Corso, Tania Cerquitelli, and Francesco Ventura. Self-tuning techniques for large scale cluster analysis on textual data collections. In *Proceedings of the Symposium on Applied Computing*, pages 771–776. ACM, 2017.

[6] Evelina Di Corso, Francesco Ventura, and Tania Cerquitelli. All in a twitter: Self-tuning strategies for a deeper understanding of a crisis tweet collection. In *IEEE BigData 2017, Boston, MA, USA* [6], pages 3722–3726.

[7] Tania Cerquitelli, Evelina Di Corso, Francesco Ventura, and Silvia Chiusano. Prompting the data transformation activities for cluster analysis on collections of documents. In *Proceedings of the 25th Italian Symposium on Advanced Database Systems, Squillace Lido (Catanzaro), Italy, June 25-29, 2017.*, page 226, 2017.

[8] Evelina Di Corso, Tania Cerquitelli, and Daniele Apiletti. Metatech: Meteorological data analysis for thermal energy characterization by means of self-learning transparent models. *Energies*, 11(6):1336, 2018.

[9] Evelina Di Corso, Tania Cerquitelli, Marco Savino Piscitelli, and Alfonso Capozzoli. Exploring energy certificates of buildings through unsupervised data mining techniques. In *Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017 IEEE International Conference on*, pages 991–998. IEEE, 2017.

[10] Tania Cerquitelli and Evelina Di Corso. Characterizing thermal energy consumption through exploratory data mining algorithms. In *EDBT/ICDT Workshops*, 2016.

[11] Tania Cerquitelli, Gianfranco Chicco, Evelina Di Corso, Francesco Ventura, Giuseppe Montesano, Mirko Armiento, Alicia Mateo González, and Andrea Veiga Santiago. Clustering-based assessment of residential consumers from hourly-metered data. In *2018 International Conference on Smart Energy Systems and Technologies (SEST)*, pages 1–6. IEEE, 2018.

[12] Tania Cerquitelli, Gianfranco Chicco, Evelina Di Corso, Francesco Ventura, Giuseppe Montesano, Anita Del Pizzo, Alicia Mateo González, and Eduardo Martin Sobrino. Discovering electricity consumption over time for residential consumers through cluster analysis. In *2018 International Conference on Development and Application Systems (DAS)*, pages 164–169. IEEE, 2018.

[13] Elena Daraio, Evelina Di Corso, Tania Cerquitelli, and Silvia Chiusano. Characterizing air-quality data through unsupervised analytics methods. In *European Conference on Advances in Databases and Information Systems*, pages 205–217. Springer, 2018.

[14] Tania Cerquitelli, Evelina Di Corso, Stefano Proto, Alfonso Capozzoli, Fabio Bellotti, Maria Giovanna Cassese, Elena Baralis, Marco Mellia, Silvia Casagrande, and Martina Tamburini. Exploring energy performance certificates through visualization. In *Proceedings of the Workshops of the EDBT/ICDT 2019 Joint Conference, EDBT/ICDT 2019, Lisbon, Portugal, March 26, 2019.*, 2019.

[15] Baoli Li and Liping Han. Distance weighted cosine similarity measure for text classification. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 611–618. Springer, 2013.

[16] Samuel Fosso Wamba, Shahriar Akter, Andrew Edwards, Geoffrey Chopin, and Denis Gnanzou. How 'big data'can make big impact: Findings from a systematic review and a longitudinal case study. *International Journal of Production Economics*, 165:234–246, 2015.

[17] Shoban Babu Sriramoju. *Introduction to big data: infrastructure and networking considerations*. Horizon Books (A Division of Ignited Minds Edutech P Ltd), 2017.

[18] M Hariesh Ramanathan. Survey of data driven medical treatment suggestion systems.

[19] Qingchen Zhang, Laurence T Yang, and Zhikui Chen. Deep computation model for unsupervised feature learning on big data. *IEEE Transactions on Services Computing*, 9(1):161–171, 2016.

[20] Pengjie Zhang, Xu Xu, and Ning Wang. The application in basketball technical actions analysis by data mining. *RISTI (Revista Iberica de Sistemas e Tecnologias de Informacao)*, (E7):348–357, 2016.

[21] Tongtao Yue, Shuangyang Li, Xianren Zhang, and Wenchuan Wang. The relationship between membrane curvature generation and clustering of anchored proteins: a computer simulation study. *Soft Matter*, 6(24):6109–6118, 2010.

[22] Suzanne Blanc, Jolley Bruce Christman, Roseann Liu, Cecily Mitchell, Eva Travers, and Katrina E Bulkley. Learning to learn from data: Benchmarks and instructional communities. *Peabody Journal of Education*, 85(2):205–225, 2010.

[23] Ganesh Kumar Venayagamoorthy. Dynamic, stochastic, computational, and scalable technologies for smart grids. *IEEE Computational Intelligence Magazine*, 6(3):22–35, 2011.

[24] Chaogan Yan and Yufeng Zang. Dparsf: a matlab toolbox for" pipeline" data analysis of resting-state fmri. *Frontiers in systems neuroscience*, 4:13, 2010.

[25] David Hakken, Maurizio Teli, and Barbara Andrews. *Beyond Capital: Values, Commons, Computing, and the Search for a Viable Future*. Routledge, 2015.

[26] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[27] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[28] Gaurangi Saxena and Siddharth Santurkar. An iterative mapreduce framework for sports-based tweet clustering. In *Proceedings of the Sixth International Conference on Computer and Communication Technology 2015*, ICCCT '15, pages 9–14, New York, NY, USA, 2015. ACM.

[29] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[30] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, Yiming Yang, et al. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA broadcast news transcription and understanding workshop*, volume 1998, pages 194–218. Citeseer, 1998.

[31] Ameni Bouaziz, Célia da Costa Pereira, Christel Dartigues Pallez, and Frédéric Precioso. Interactive generic learning method (iglm): A new approach to interactive short text classification. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16, pages 847–852, New York, NY, USA, 2016. ACM.

[32] Timo Duchrow, Timur Shtatland, Daniel Guettler, Misha Pivovarov, Stefan Kramer, and Ralph Weissleder. Enhancing navigation in biomedical databases by community voting and database-driven text classification. *BMC Bioinformatics*, 10:317, 2009.

[33] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

[34] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[35] Oskar Gross, Antoine Doucet, and Hannu Toivonen. Language-independent multi-document text summarization with document-specific word associations. In *31st Annual ACM Symposium on Applied Computing (ACM SAC)*, Pisa, Italy, 2016. ACM, ACM.

[36] Hasan M. Jamil and Hosagrahar V. Jagadish. A structured query model for the deep relational web. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1679–1682, New York, NY, USA, 2015. ACM.

[37] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. In *OSDI'04*, pages 10–10, 2004.

[38] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *NSDI'12*, pages 2–2, 2012.

[39] Pinar Alper, Khalid Belhajjame, Carole A Goble, and Pinar Karagoz. Enhancing and abstracting scientific workflow provenance for data publishing. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 313–318. ACM, 2013.

[40] Jitendra Nath Singh and Sanjay Kumar Dwivedi. A comparative study on approaches of vector space model in information retrieval. In *International Conference of Reliability, Infocom Technologies and Optimization*, 2013.

[41] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[42] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. In *Advances in neural information processing systems*, pages 601–608, 2002.

[43] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[44] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[45] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[46] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[47] Matthew R Hallowell and John A Gambatese. Construction safety risk mitigation. *Journal of Construction Engineering and Management*, 135(12):1316–1323, 2009.

[48] Nirup M Menon, Byungtae Lee, and Leslie Eldenburg. Productivity of information systems in the healthcare industry. *Information Systems Research*, 11(1):83–92, 2000.

[49] Adam Bossler and Thomas J Holt. *Cybercrime in progress: Theory and prevention of technology-enabled offenses*. Routledge, 2015.

[50] Kurt Joseph, Benjamin Knott, Robert Bushey, and Theodore Pasquale. Method for identifying and prioritizing customer care automation, February 2 2006. US Patent App. 10/901,926.

[51] David L Bauer, Keith R McFarlane, Andrew Derek Flockhart, Lucinda M Sanders, Gary S King, Darryl J Maxwell, Steve R Russell, Robert Alan Stewart, and Wendy S Cook. Integrated work management engine for customer care in a communication system, February 10 2004. US Patent 6,690,788.

[52] Ion Androutsopoulos, John Koutsias, Konstantinos V Chandrinos, George Paliouras, and Constantine D Spyropoulos. An evaluation of naive bayesian anti-spam filtering. *arXiv preprint cs/0006013*, 2000.

[53] Le Zhang, Jingbo Zhu, and Tianshun Yao. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4):243–269, 2004.

[54] Ralph I Allison and Kenneth P Uhl. Influence of beer brand identification on taste perception. *Journal of Marketing Research*, pages 36–39, 1964.

[55] Rudolf R Sinkovics, Elfriede Penz, and Pervez N Ghauri. Analysing textual data in international marketing research. *Qualitative Market Research: An International Journal*, 8(1):9–38, 2005.

[56] Andrei Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. A semantic approach to contextual advertising. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 559–566. ACM, 2007.

[57] Prabin Lama. Clustering system based on text mining using the k-means algorithm. *Turku University of Applied Sciences. Finlandia*, 2013.

[58] Charu Aggarwal and Chengxiang Zhai. A survey of text clustering algorithms. *Mining Text Data*, 08 2012.

[59] Douglass R Cutting, David R Karger, Jan O Pedersen, and John W Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–329. ACM, 1992.

[60] Hinrich Schütze and Craig Silverstein. Projections for efficient document clustering. 1997.

[61] L Douglas Bakeryz and Andrew Kachites McCallumyz. Distributional clustering of words for text classification. In *Proceedings of SIGIR*. Citeseer, 1998.

[62] Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, and Yoad Winter. On feature distributional clustering for text categorization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 146–153. ACM, 2001.

[63] Kamal Nigam, Andrew McCallum, Sebastian Thrun, Tom Mitchell, et al. Learning to classify text from labeled and unlabeled documents. *AAAI/IAAI*, 792:6, 1998.

[64] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[65] M Sridevi, R Rajeshwara Rao, and M Varaprasad Rao. A survey on recommender system. *International Journal of Computer Science and Information Security*, 14(5):265, 2016.

[66] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Bosagh Zadeh. Wtf: The who-to-follow system at twitter. In *Proceedings of the 22nd international conference on World Wide Web WWW*, 2013.

[67] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006.

[68] Ricardo Baeza-Yates, Berthier de Araújo Neto Ribeiro, et al. *Modern information retrieval*. New York: ACM Press; Harlow, England: Addison-Wesley,, 2011.

[69] Susan T Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers*, 23(2):229–236, 1991.

[70] Preslav Nakov, Antonia Popova, and Plamen Mateev. Weight functions impact on LSA performance. In *EuroConference RANLP'2001 (Recent Advances in NLP*, pages 187–193, 2001.

[71] Anindya Roy and Sudipto Banerjee. *Linear algebra and matrix analysis for statistics*. Chapman and Hall/CRC, 2014.

[72] LIII KPFRS. On lines and planes of closest fit to systems of points in space. In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (SIGMOD)*, 1901.

[73] Susan T Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.

[74] Ivan Markovsky and KONSTANTIN Usevich. *Low rank approximation*. Springer, 2012.

[75] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[76] Michael W Berry and Malu Castellanos. Survey of text mining. *Computing Reviews*, 45(9):548, 2004.

[77] David G Underhill, Luke K McDowell, David J Marchette, and Jeffrey L Solka. Enhancing text analysis via dimensionality reduction. In *Information Reuse and Integration, 2007. IRI 2007. IEEE International Conference on*, pages 348–353. IEEE, 2007.

[78] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[79] Russ Albright. Taming text with the svd. *SAS Institute Inc*, 2004.

[80] Giuseppe C Calafiore, Laurent El Ghaoui, Alessandro Preziosi, and Luigi Russo. Topic analysis in news via sparse learning: a case study on the 2016 us presidential elections. *IFAC-PapersOnLine*, 50(1):13593–13598, 2017.

[81] Youwei Zhang and Laurent E Ghaoui. Large-scale sparse principal component analysis with application to text data. In *Advances in Neural Information Processing Systems*, pages 532–539, 2011.

[82] Laurent El Ghaoui, Vu Pham, Guan-Cheng Li, Viet-An Duong, Ashok Srivastava, and Kanishka Bhaduri. Understanding large text corpora via sparse machine learning. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 6(3):221–242, 2013.

[83] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[84] Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. Auto-encoder bottleneck features using deep belief networks. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4153–4156. IEEE, 2012.

[85] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.

[86] Yuan-chu Cheng, Wei-Min Qi, and Wei-you Cai. Dynamic properties of elman and modified elman neural network. In *Proceedings. International Conference on Machine Learning and Cybernetics*, volume 2, pages 637–640. IEEE, 2002.

[87] Andrea Baraldi and Palma Blonda. A survey of fuzzy clustering algorithms for pattern recognition. i. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(6):778–785, 1999.

[88] Anthony McGregor, Mark Hall, Perry Lorier, and James Brunskill. Flow clustering using machine learning techniques. In *International workshop on passive and active network measurement*, pages 205–214. Springer, 2004.

[89] Francesco Ventura, Tania Cerquitelli, and Francesco Giacalone. Black-box model explained through an assessment of its interpretable features. In *European Conference on Advances in Databases and Information Systems*, pages 138–149. Springer, 2018.

[90] James D Foley, Foley Dan Van, Andries Van Dam, Steven K Feiner, John F Hughes, J HUGHES, and EDWARD ANGEL. *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional, 1996.

[91] Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM Journal on Computing*, 33(6):1417–1440, 2004.

[92] Kevin de Queiroz and David A Good. Phenetic clustering in biology: a critique. *The Quarterly Review of Biology*, 72(1):3–30, 1997.

[93] Ioannis Iliopoulos, Anton J Enright, and Christos A Ouzounis. Textquest: document clustering of medline abstracts for concept discovery in molecular biology. In *Biocomputing 2001*, pages 384–395. World Scientific, 2000.

[94] Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD explorations newsletter*, 4(1):65–75, 2002.

[95] Min Zhang and Jian Yu. Fuzzy partitional clustering algorithms [j]. *Journal of Software*, 6:007, 2004.

[96] Ying Zhao and George Karypis. Comparison of agglomerative and partitional document clustering algorithms. Technical report, MINNESOTA UNIV MINNEAPOLIS DEPT OF COMPUTER SCIENCE, 2002.

[97] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[98] Xinyu Wang, Julien Ah-Pine, and Jérôme Darmont. Shcoclust, a scalable similarity-based hierarchical co-clustering method and its application to textual collections. In *Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[99] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[100] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

[101] Kuan-Ching Li, Hai Jiang, Laurence T Yang, and Alfredo Cuzzocrea. *Big data: Algorithms, analytics, and applications*. CRC Press, 2015.

[102] Sanjay Joshi and Tien-Chien Chang. Graph-based heuristics for recognition of machined features from a 3d solid model. *Computer-Aided Design*, 20(2):58–66, 1988.

[103] Erez Hartuv and Ron Shamir. A clustering algorithm based on graph connectivity. *Information processing letters*, 76(4-6):175–181, 2000.

[104] B.-H. Juang and L.R. Rabiner. The segmental k-means algorithm for estimating parameters of hidden markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(9):1639–1641, Sep 1990.

[105] T. Pang-Ning, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.

[106] Dan Oneata. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty*, pages 1–7, 1999.

[107] Florent Monay and Daniel Gatica-Perez. Plsa-based image auto-annotation: constraining the latent space. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 348–351. ACM, 2004.

[108] Josef Sivic, Bryan C Russell, Alexei A Efros, Andrew Zisserman, and William T Freeman. Discovering object categories in image collections. 2005.

[109] Rainer Lienhart, Stefan Romberg, and Eva Hörster. Multilayer plsa for multimodal image retrieval. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, page 9. ACM, 2009.

[110] Matt Hoffman, D Blei, and Perry R Cook. Finding latent sources in recorded music with a shift-invariant hdp. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 438–444, 2009.

[111] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[112] Roger B Bradford. An empirical study of required dimensionality for large-scale latent semantic indexing applications. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 153–162. ACM, 2008.

[113] X Hu, Z Cai, D Franceschetti, P Penumatsa, AC Graesser, MM Louwerse, DS McNamara, Tutoring Research Group, et al. Lsa: First dimension and dimensional weighting. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 25, 2003.

[114] Richard Cangelosi and Alain Goriely. Component retention in principal component analysis with application to cdna microarray data. *Biology direct*, 2(1):2, 2007.

[115] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

[116] B-H Juang and Lawrence R Rabiner. The segmental k-means algorithm for estimating parameters of hidden markov models. *IEEE Transactions on acoustics, speech, and signal Processing*, 38(9):1639–1641, 1990.

[117] Weizhong Zhao, James J Chen, Roger Perkins, Zhichao Liu, Weigong Ge, Yijun Ding, and Wen Zou. A heuristic approach to determine an appropriate number of topics in topic modeling. *BMC bioinformatics*, 16(13):S8, 2015.

[118] Wen Zhang, Yangbo Cui, and Taketoshi Yoshida. En-lda: An novel approach to automatic bug report assignment with entropy optimized latent dirichlet allocation. *Entropy*, 19(5):173, 2017.

[119] Ah-Hwee Tan et al. Text mining: The state of the art and the challenges. In *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, volume 8, pages 65–70. sn, 1999.

[120] John Stasko, Carsten Görg, and Zhicheng Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization*, 7(2):118–132, 2008.

[121] Andy Kirk. *Data visualisation: a handbook for data driven design*. Sage, 2016.

[122] W Bradford Paley. Textarc: Showing word frequency and distribution in text. In *Poster presented at IEEE Symposium on Information Visualization*, volume 2002, 2002.

[123] Roel Popping. Knowledge graphs and network text analysis. *Social Science Information*, 42(1):91–106, 2003.

[124] Roel Popping. *Computer-assisted text analysis*. Sage, 2000.

[125] Allan Heydon and Marc Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, 1999.

[126] Sholom M Weiss, Nitin Indurkhya, and Tong Zhang. *Fundamentals of predictive text mining*. Springer, 2015.

[127] Tania Cerquitelli, Evelina Di Corso, Francesco Ventura, and Silvia Chiusano. Prompting the data transformation activities for cluster analysis on collections of documents. In *Proceedings of SEBD 2017*, pages 226–234, 2017.

[128] Pengtao Xie and Eric P Xing. Integrating document clustering and topic modeling. *arXiv preprint arXiv:1309.6874*, 2013.

[129] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2017.

[130] Gang Qian, Shamik Sural, Yuelong Gu, and Sakti Pramanik. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1232–1237. ACM, 2004.

[131] John Hopcroft and Ravi Kannan. Computer science theory for the information age. 2012.

[132] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[133] Matthew Hoffman, Francis R. Bach, and David M. Blei. Online learning for latent dirichlet allocation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 856–864. Curran Associates, Inc., 2010.

[134] Ellen Spertus, Mehran Sahami, and Orkut Buyukkokten. Evaluating similarity measures: a large-scale study in the orkut social network. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 678–684. ACM, 2005.

[135] W Ben Towne, Carolyn Penstein Rosé, and James D Herbsleb. Measuring similarity similarly: Lda and human perception. *ACM TIST*, 8(1):7–1, 2016.

[136] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proceedings of the VLDB Endowment*, 5(7):622–633, 2012.

[137] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael J. Franklin, Reza Zadeh, Matei Zaharia, and Ameet Talwalkar. Mllib: Machine learning in apache spark. *J. Mach. Learn. Res.*, 17(1):1235–1241, January 2016.

[138] Andrea Vattani. K-means requires exponentially many iterations even in the plane. *Discrete & Computational Geometry*, 45(4):596–616, 2011.

[139] David Arthur and Sergei Vassilvitskii. How slow is the k-means method? In *Symposium on Computational Geometry*, volume 6, pages 1–10, 2006.

[140] Michael Shindler, Alex Wong, and Adam W Meyerson. Fast and accurate k-means for large datasets. In *Advances in neural information processing systems*, pages 2375–2383, 2011.

[141] David Sontag and Dan Roy. Complexity of inference in latent dirichlet allocation. In *Advances in neural information processing systems*, pages 1008–1016, 2011.

[142] Jia Zeng, Zhi-Qiang Liu, and Xiao-Qin Cao. Fast online em for big topic modeling. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):675–688, 2016.

[143] Eva Hörster, Rainer Lienhart, and Malcolm Slaney. Image retrieval on large-scale image databases. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 17–24. ACM, 2007.

[144] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[145] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[146] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864, 2003.

[147] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[148] Jason Chuang, Christopher D Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the international working conference on advanced visual interfaces*, pages 74–77. ACM, 2012.

[149] Florian Heimerl, Steffen Lohmann, Simon Lange, and Thomas Ertl. Word cloud explorer: Text analytics based on word clouds. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 1833–1842. IEEE, 2014.

[150] Kevin R Canini, Bongwon Suh, and Peter L Pirolli. Finding credible information sources in social networks based on content and social structure. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 1–8. IEEE, 2011.

[151] Jonathan L Gross, Jay Yellen, and Ping Zhang. *Handbook of graph theory*. Chapman and Hall/CRC, 2013.

[152] Béla Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 2013.

[153] Derek Greene and Pádraig Cunningham. Producing a unified graph representation from multiple social network views. In *Proceedings of the 5th annual ACM web science conference*, pages 118–121. ACM, 2013.

[154] Rada Mihalcea and Dragomir Radev. *Graph-based natural language processing and information retrieval*. Cambridge university press, 2011.

[155] Quoc-Dinh Truong Inglebert. Community retrieval and visualization in large graphs.

[156] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[157] Claudia Diamantini, Laura Genga, and Domenico Potena. Esub: Exploration of subgraphs. *Proceedings of the BPM demo session*, pages 70–74, 2015.

[158] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[159] Reinhard Diestel. Graph theory, electronic edition 2000 ed, 2000.

[160] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.

[161] Rakesh Agrawal, Tomasz Imilienski, and Arum Swami. Mining association rules between sets of items in large databases. *In SIGMOD'93 , Washington DC*, May 1993.

[162] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[163] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[164] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

[165] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[166] Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web*, pages 585–594. ACM, 2006.

[167] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. AcM, 2010.

[168] Abdul Manan Koli, Muqeem Ahmed, and Jatinder Manhas. An empirical study on potential and risks of twitter data for predicting election outcomes. In *Emerging Trends in Expert Applications and Security*, pages 725–731. Springer, 2019.

[169] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *International AAAI Conference on Web and Social Media*, 2014.

[170] Jose A Miñarro-Giménez, Markus Kreuzthaler, and Stefan Schulz. Knowledge extraction from medline by combining clustering with natural language processing. In *AMIA Annual Symposium Proceedings*, volume 2015, page 915. American Medical Informatics Association, 2015.

[171] Jamie O'Keeffe, John Willinsky, and Lauren Maggio. Public access and use of health research: an exploratory study of the national institutes of health (nih) public access policy using interviews and surveys of health personnel. *Journal of medical Internet research*, 13(4), 2011.

[172] Iman Saleh and Neamat El-Tazi. Automatic organization of semantically related tags using topic modelling. In *Advances in Databases and Information Systems*, pages 235–245. Springer, 2017.

[173] Justin Wood, Patrick Tan, Wei Wang, and Corey Arnold. Source-lda: Enhancing probabilistic topic models using prior knowledge sources. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*, pages 411–422. IEEE, 2017.

[174] Franca Debole, Fabrizio Sebastiani, and Via Giuseppe Moruzzi. An analysis of the relative difficulty of reuters-21578 subsets. In *LREC*, 2004.

[175] Catarina Silva and Bernadete Ribeiro. *Inductive inference for large scale text classification: kernel approaches and techniques*, volume 255. Springer, 2009.

[176] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[177] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[178] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[179] George Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[180] Evelina Di Corso, Stefano Proto, Tania Cerquitelli, and Silvia Chiusano. Towards automated visualisation of scientific literature. In *European Conference on Advances in Databases and Information Systems*. Springer, 2019.